RL-TR-91-417
Final Technical Report
December 1991

(2)

FIXED TARGET COMPARATIVE ANALYSIS

Control Data Corporation

Michael E. Murphy, Brent Rickenback, Harlan Paetznick

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-91-417 has been reviewed and is approved for publication.

APPROVED: Charles R. Walling

CHARLES R. WALLING, MSgt, USAF
Project Engineer

FOR THE COMMANDER: Thadeus Domurat

THADEUS J. DOMURAT
Technical Director
Intelligence & Reconnaissance Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1991	Final Dec 88 - Oct 90

4. TITLE AND SUBTITLE
FIXED TARGET COMPARATIVE ANALYSIS

5. FUNDING NUMBERS
C - F30602-88-C-0075
PE - 63260F
PR - 3480
TA - 00
WU - 26

6. AUTHOR(S)
Michael E. Murphy, Brent Rickenbach, Harlan Paetznick

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Control Data Corporation
2800 E. Old Shakopee Road
Minneapolis MN 55440

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory (IRRE)
Griffiss AFB NY 13441-5700

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
RL-TR-91-417

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: Charles R. Walling, MSgt, USAF/IRRE/(315) 330-4592

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
This effort was conducted to analyze and automate the process of performing change detection on fixed target sites. It automates the process of comparative analysis between images of the same target area collected at different times. It combines new digital processing technologies with state-of-the-art image exploitation techniques on a user friendly workstation. FTCA utilizes a 3-D reference data base describing the location and size of all targets/objects within the area. It uses multiple sensor fusion of imagery and 3-D frame models to detect changes and cues the analyst to these changes for further analysis.

14. SUBJECT TERMS
3-D, Target Area Changes, Comparative Analysis, 3 Dimensional, Imagery Analysis, Change Detection

15. NUMBER OF PAGES
120

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std. Z39-18
298-102

TABLE OF CONTENTS

LIST OF FIGURES

1.0 INTRODUCTION AND SUMMARY

This is the final technical report for the Fixed Target Comparative Analysis Contractual Effort. The technical effort during this 21 month contract was performed by the Research and Technology Operations (RTO) Organization within Government Systems Group of Control Data Corporation. RTO is directed by Mr. Bruce Colton. Sergeant Charles Walling was the Program Engineer at Rome Air Development Center and his insight and guidance are gratefully acknowledged. Michael Murphy was the program manager for Control Data. He was assisted at various points during the development by Dr. Hwa-Chuang Chien, Scott Devitt, Charles Grosch, Harlan Paetznick, James Polzin, Brent Rickenbach, and Thomas Rosenthal. In the remainder of section one we will provide background on the requirement, present the selected approach, highlight the delivered system's functionality, and provide general conclusions on the development.

1.1 Statement of Requirement

This effort was intended to analyze and automate the process of performing comparative analysis for fixed target sites. For the most part this is currently a manual operation performed by trained Image Analysts. The analysts rely upon hard-copy images covering the target area, intelligence database textual printouts describing the target area, and previously performed readouts as a basis for generating new intelligence reports describing the current status of the target site. Each of these sources is analyzed in turn by the analysts, relying upon their knowledge of specific targets and past experience in order to construct the target area report. Occasionally collateral data like maps or target area graphics (e.g., ATTGs) are available and the Image Analysts can use these during their analysis. Thus the demanding and time-consuming nature of the job is clear.

If the target area happens to be in a high priority class, it is reviewed frequently. Typically the same analyst is given responsibility for the area and becomes familiar with the topography and the level of activity that can be considered normal. If the target area is of lower priority, the analyst will not be familiar with the terrain or expected activity levels in the area. In these cases, the preparation time for the analyst is increased, since more time will be spent collecting any available collateral information which might help in interpreting the new image. Additional time is also spent getting oriented once the imagery is obtained, since the contents of the textual database must be assimilated and then related to the new image.

The Fixed Target Comparative Analysis Effort is intended to address this last category of target area, that is lower priority target areas reviewed intermittently. Four classes of fixed targets were selected for development and testing. Included

were barracks, missile sites, airfields, and storage areas. The goals of the effort as stated in the Statement of Work are:

- Analyze, design, modify/develop, install, integrate, test, and document software that will allow for the automatic detection of man-made objects at fixed military-related facilities.
- Demonstrate near real-time performance.
- Demonstrate potential improvement in managing exploitation resources through automation.
- Document Results and Performance.

The Statement of Work further refines these requirements providing further breakdown and clarification of the comparative analysis function. The reader is referred to the Statement of Work for those details as needed.

1.2 Approach

Control Data with RADC's concurrence decided to pursue an approach based upon constructing three dimensional models of all objects of interest within the target site. Control Data could propose this approach based upon previously developed site modeling capabilities. Thus minimal new software development was required to produce the 3D models to support evaluation of this 3D model-supported exploitation approach. The comparative analysis techniques that were developed exploited the fact that the 3D site model data base was available during the automatic processing. This opened up a new range of processing strategies during the automated image comparison function.

In addition, the existence of the 3D site models allows for the development of an easy-to-use analyst interface for rapidly identifying objects of interest within the target site. The existence of the 3D models, and more importantly the static knowledge structure which they imply, provides an extensible focusing mechanism. Information about the target site can be incrementally captured and recorded. More importantly, the information recorded in 3D site model coordinates can be projected as graphical entities over any image covering the target site, thus facilitating rapid understanding of the important objects within each site.

After formulating this high level approach, a visit to the 480th RTG/INPOE at Langley Air Force was undertaken to obtain better insight into the working environment for an eventual system and to present an outline of the proposed approach to obtain reactions and feedback from potential users. The meeting was

very informative and a beneficial interchange between Control Data and Air Force personnel took place. New insights into the working environment, more substantive details of the reporting process, and new insights into the sorts of objects that are important were obtained.

With the goal of developing a 3D model-supported exploitation system and knowledge of the intended working environment for such a system, Control Data and RADC revised the approach to include delivery of a fully functional workstation implementation of the FTCA System. The Silicon Graphics Iris 4D workstation was selected because of it's superior performance in the area of 3D object presentation and manipulation. This selection was also in concert with RADC's long term plan for building an open systems architecture via connecting multiple vendors equipment within their Image Processing Laboratory(IPL). In addition, the workstation uses the Unix Operating System, the implementation would be coded using the C language, and support for X Windows is available thereby providing software implementation consistent with existing systems within RADC's IPL.

1.3 Developed Functionality

Control Data developed a stand alone system for assisting Image Analysts with the comparative analysis task. We believe the delivered system represents a vision-of-the-future in terms of analyst workstation requirements because of the advanced concepts underlying the system design and resultant implementation for a number of reasons. The windowing system provided with the FTCA system represents a novel approach to the presentation of fused information which is flexible, sensible, and easy for the analyst to use. In addition, it encourages incorporation of new techniques as they are developed and facilitates their integration. The FTCA System makes explicit use of geometric information about the objects of interest within the target area. The system provides an extensible, tailored image analyst toolset which can be accessed at any time by the analyst. The automated and semi-automated comparative analysis techniques provided by the FTCA system are innovative in the sense that they begin to incorporate the kinds of geometric and reflective understanding which is essential to address the subtleties of the comparative analysis task. Finally the entire system is delivered on a powerful, compact, multi-purpose workstation.

The Fixed Target Comparative Analysis Task like many other intelligence collection tasks, is one in which multiple sources of information must be analyzed in concert in order for an Image Analyst to formulate a sufficient level of understanding to produce data for the target's EEIs. The types of information the analyst needs to view are varied in form and format. Included are images, text, graphics, cues, and other forms of collateral data (e.g., charts, ATTGs and NBRGs). Control Data, using

Research and Development Funds developed a windowing system to address this issue.

This capability termed the Geo-Located Multi-source eXploitation Windowing System (GLMX), was provided (on a no cost license basis) to RADC along with all the applications comprising the FTCA system. GLMX fulfills an integrating function in the sense that it provides an environment into which all applications functions can be more or less "pasted" after they have each been independently developed. GLMX provides a number of services which make it easy for developers to produce application programs which can coexist with other applications and mutually support each other by performing their independent functions for the image analyst in a consistent fashion.

GLMX also provides an information fusion function. The independently developed functions, each of which provides a service or supplies some form of information for the analyst, all become "stackable" within a GLMX window. They mutually support and build upon each other because they present their information in a fused fashion as directed by GLMX. Thus the analyst is able to stack applications, each providing it's form of information, in a single window in any desired order. Furthermore the analyst has the freedom to turn the application on and off at any time. Thus the analyst can effectively "program" the window to contain exactly the information necessary to satisfy the current exploitation task.

A portion of the delivered FTCA system functions constitute an image analyst's toolset which is intended to assist in performing various aspects of the comparative analysis task. Examples include utilities to perform mensuration, multi-image cursor tracking, image enhancement, image statistics presentation, image collection parameters display, site model presentation, etc. All of these tools are easily accessed by the analyst on an as-needed basis.

The FTCA system was developed around the concept of having and utilizing a 3 dimensional target area site model. The site model provides the geometric specification of the objects of interest within the target area. Additional characteristics of the objects (e.g. surface material(s) of the objects) can be represented as part of the site model as well. The site model provides a common coordinate system and therefore a standard information referencing mechanism. In addition, it serves as an extensible knowledge structure which holds and organizes relevant information about the target area.

Multiple automatic comparative analysis techniques are provided by the FTCA system. The comparison techniques use various combinations of the site model, a previously collected reference image, and a newly collected mission image. As described in latter sections of the final report, the existence of the site model

facilitates a rich new class of automatic comparative analysis strategies which are much more tolerant of widely varying image collection geometry. Included are techniques based upon searching for visible edges caused by expected surface discontinuities, predicting, detecting, and comparing object-induced shadows, and multiple change detection techniques based upon comparing image intensity data between image pairs after precise alignment via known scene geometry provided by the site model.

1.4 General Conclusions

After completing the development and system installation at RADC the following conclusions can be offered.

- The advantage of constructing, maintaining, and incorporating a site model into the exploitation process for fixed target areas is significant and justified on a cost-to-produce versus derived-benefit basis.

- The GLMX Windowing System is well suited to image exploitation tasks and offers a benchmark for exploitation workstation user interface concepts.

- The model supported exploitation concept, especially for fixed target sites, is one which has tremendous appeal and applicability over a wide range of government organizations dealing with intelligence data collection and subsequent usage.

- The Silicon Graphics Workstation, although relatively inexpensive, appears to be an excellent platform for image exploitation activities as well as support functions related to intelligence data collection, handling, and presentation consistent with all applications investigated for FTCA.

- Site modeling technology and model-supported exploitation technology is sufficiently developed to warrant consideration in terms of potential inclusion in a production system scenario. In the near term, the technology could be used in a hard-copy product generation and dissemination mode in support of explicit exploitation tasks. In parallel, softcopy exploitation approaches and possibilities at selected sites should be explored.

In the remainder of this final technical report we will provide the supporting data for these conclusions.

2.0 MODEL-SUPPORTED EXPLOITATION RATIONALE/REQUIREMENTS

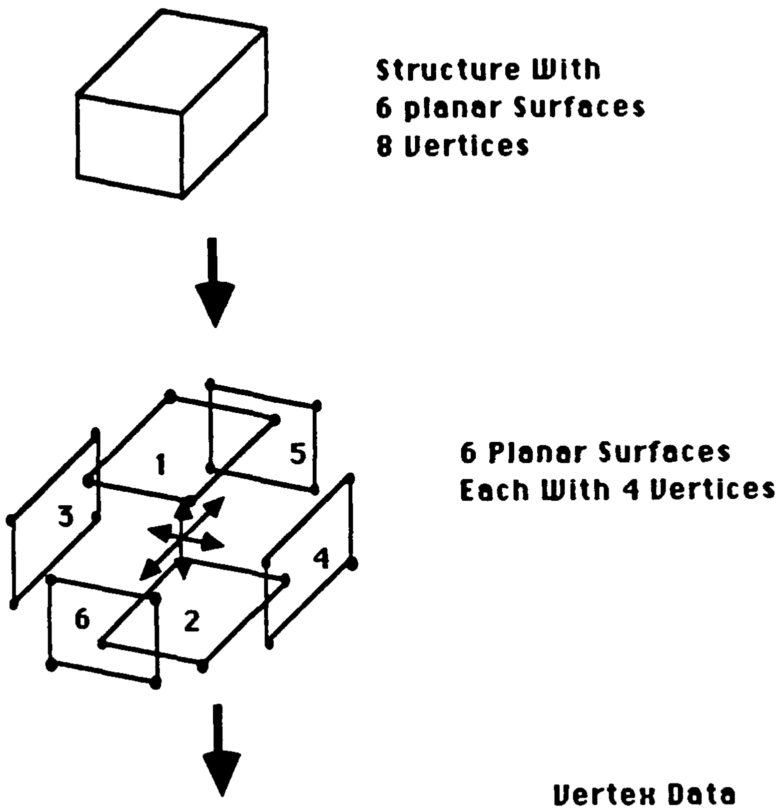
In this section we will discuss the 3 dimensional target area site models from the standpoint of requirements and usage rationale. The discussion will be focussed in 3 areas. An overview of the content of the site models used for the FTCA investigations will be provided as well as an estimate of their adequacy. This will be followed by a discussion of the potential for rapid site model production in support of systems like FTCA and/or others applications. Finally an overview of the potential benefits that site modeling and model-supported exploitation facilitate will be provided.

2.1 Site Model Content

It should be understood that the site models produced and utilized for the FTCA investigations were generated by Control Data only to serve in a concept demonstration/validation role. To our knowledge no specification which formally defines the required content and fidelity of a 3D target area site model exists. Because of time constraints the site models produced for the FTCA effort were necessarily simple. Nevertheless they were certainly adequate for conveying the site model concept, and more importantly for developing, evaluating, and demonstrating model-supported exploitation concepts.

All of the site models produced for the FTCA investigation are maintained in a hierarchical data base format. The actual data consists of a series of records defining structures (objects) which are in turn composed of a number of planar surfaces, each of which is bounded by a number of vertices. Figure 2-1 depicts the organization and prototypical data for a structure (object) within a 3D site model data base. The structure is composed of 6 planar surfaces each with 4 vertices which define their boundaries in 3 space. The surface data describes the location and attribute data (surface material, for example) of each surface of the structure. Vertices are simply locations in 3 space.

The site models used for FTCA contain a fairly limited set of object types including single planar surfaces, gable-roof structures, hip-roof structures, right prisms, cylinders, cones, and general prisms for example. The surface material fields were not filled so only the geometric data was present.



Surface Data

ID	D	N1	N2	N3	a1	nv	v1	v2	v3	v4	•••	vn
S1												
S2												
S3												
S4												
S5												
S6												

Vertex Data

ID	X	Y	Z
V1			
V2			
V3			
V4			
V5			
V6			
V7			
V8			

Figure 2-1. Sample site model object data structure.

2.1.1 Future Site Model Content Requirements

The site models used within FTCA were adequate for concept evaluation and demonstration purposes. Clearly much could be and is being done in the area of site model data base extension and refinement. Some of this will be driven by the applications which make use of site model data. We believe these are still being conceived and developed so the definition of a site model will be changing often in the short term. The following suggestions are the result of the FTCA study and extensions which we believe make sense for other applications.

The site models used within FTCA consist of a single planar surface to define the ground plane (surface) upon which other structures reside. This should be modified to accept a faceted surface as the extent of the site model grows. Provision for DTED usage should also be included.

The fidelity of the site models for individual objects should be extended. Provisions for doors, windows, special reflective locations, etc. should be included. Along with this extension, the spatial relationships between and among site model components needs further definition.

Object attribute data should be extended to include a textual descriptive field for the image analyst. Included would be descriptive data important to an intelligence analyst. This data could be used for example, by an extended "preview" function within FTCA which would selectively retrieve and display the data for the analyst on an object-by-object basis.

Expansion of the surface material, surface rendering capabilities should be included. Sensor simulation applications are becoming more useful and their requirements more well defined. Those requirements should be incorporated into the site model definition. This is also required from the model-supported exploitation point of view. To develop techniques which utilize surface material, roughness, reflectivity, etc. as stored in a site model, and then predict and confirm those finding with specific image processing techniques obviously requires the initial site model definition.

Image perspective transformation support data provisions may also be included. It may make sense to record a pointer to the best image of each individual surface in the site model and/or alternatively to store a simple iconic record for each surface to facilitate synthetic image generation.

2.2 Site Model Production

In order for model-supported exploitation to become feasible it is clear a rapid site modeling capability must be in place. In the case of FTCA, Control Data developed the site models using a combination of site modeling technology developed in support of reference scene preparation for the Cruise Missile Advanced Guidance (CMAG) Program and a prototype rapid modeling software package developed using IR&D funds. Each site model required approximately 8 hours to produce. We believe that time must be reduced to less than 1 hour.

There are 2 major aspects to accurate site modeling. The first is camera modeling which is followed by the site model generation process. The camera model for an

image defines the transformation from scene space (3D) to image space (2D). This equates to defining the sensor position and attitude with respect to a fixed 3 space coordinate system. This can be more or less complicated depending upon the dynamics of the sensor during the image formation process. The actual process of defining a camera model also depends upon the amount of information accompanying the image in terms of defining sensor dynamics and collection geometry during image formation.

Typically we are only supplied with a pair (sometimes more) of images covering a site with no other accompanying information. This need not be a stereo pair but the images should be collected with reasonable geometric separation. We use these two images in concert and define a local rectangular 3 space coordinate system to record site model locations. Control Data has developed what we refer to as the line-preserving camera modeling [17] software package to produce a camera model for each image.

After the camera model for each image is available, the site model can be produced. This is an interactive, graphically-driven process. The human's ability to recognize and categorize objects is merged with the computers ability to manipulate and present data in a fashion which allows rapid placement and parameter adjustment of object primitives.

A rapid site modeling capability is currently in development at Control Data under RADC sponsorship. The Advanced Reference Scene Products (ARSP) development will provide a workstation implementation for rapid site model production. This development is scheduled for delivery in mid 1991.

2.3 Site Model Benefits

The advantages of having a site model during the exploitation process are significant. For this discussion we will put them into 1 of three categories as follows:

- offers low risk approach for substantially improved productivity
- provides an extensible knowledge structure for the target area
- offers improved potential for nearly automated exploitation

Site models provide the potential for significant productivity enhancement. Even the relatively simple process of projecting the wire frame depictions of the site model over the latest mission image offers an excellent low risk cueing approach for the Image Analyst. Model-supported exploitation and the productivity enhancement tools which become possible given a site model, are best utilized in a softcopy exploitation environment. Unfortunately any significant level of softcopy

exploitation is still in the future. In the short term it may be possible to further evaluate improved productivity via model-supported exploitation by turning some of the softcopy exploitation potential into standard products (hardcopy) that could be generated early in the production stream (e.g., image with wire frame overlay, solid view, etc.) and distributed to organizations performing the various exploitation tasks.

The formal definition, production, and usage of a site model provides a repository for capturing, organizing, and distributing target area information. In this sense it can be considered an extensible knowledge structure that changes over time to track the target areas status. This coupled with the visualization capabilities of todays relatively inexpensive workstations, offers the potential for rapid presentation and assimilation of vast knowledge about a target area by the Image Analyst.

The production and subsequent use of a site model to enable more fully automated exploitation is an additional key advantage. The existence of the site model offers the basis for developing additional strategies for automating image exploitation tasks. This report describes a number of examples in section 4. We believe there are many more. In addition, the site model facilitates the development of specialized tools to assist the Image Analyst even in a semi-automated implementation.

3.0 IMPLEMENTATION OVERVIEW

This section will provide an overview of the key elements of the delivered implementation. Included are the GLMX Windowing System, Image Analyst Toolset, and Model Supported Exploitation.

3.1 Geo-Located Multi-source Exploitation Windowing System

We believe the user interface and information fusion approach provided by the GLMX Windowing System is unique. Since the GLMX windowing system allows multiple information sources to be effectively "stacked" on top of each other in a single display window, the system allows very efficient use of a limited display resource. The analyst is allowed to select/deselect any sequence of information sources by clicking on the corresponding GLMX tab. Thus the analyst is in complete control of the amount and order of information being presented at all times.

Furthermore since each piece of information is being presented by an independently executing program, it's use can be tailored for the analyst independent of the presentation of any other piece of information. In addition other individual applications like enhancement, mensuration, collection information presentation, etc. can also be added to the GLMX stack at any time and selectively applied to the information source currently under review by the analyst. All of the manipulations take place in an intuitively obvious, user friendly manner.

GLMX Windows are used as the underlying methodology for information fusion within the FTCA System. These windows are an extension of the 4Sight windowing system provided by SGI for use on their IRIS 4D workstations. Thus both 4Sight and GLMX Windows run on the NeWS server. NeWS is an extended Postscript interpreter. Both 4Sight and GLMX are both primarily written in Postscript with a small amount of C interface code. The SGI Graphic Library (GL) functions are supported by both GLMX and 4Sight windows. Currently GLMX Windows run only on the SGI systems to leverage the advantages of the Geometry Pipeline for superior 3D operations on these machines. Since most of the code is Postscript they could be made to run on other systems (e.g. SUN) with minor modifications.

The GLMX Windowing System provides a number of standard services which are tailored to the exploitation of image data. In the case of FTCA, they further exploit the fact that a coordinate system tied to the actual scene is available . Figure 3-1 depicts a GLMX window.

GLMX Windows

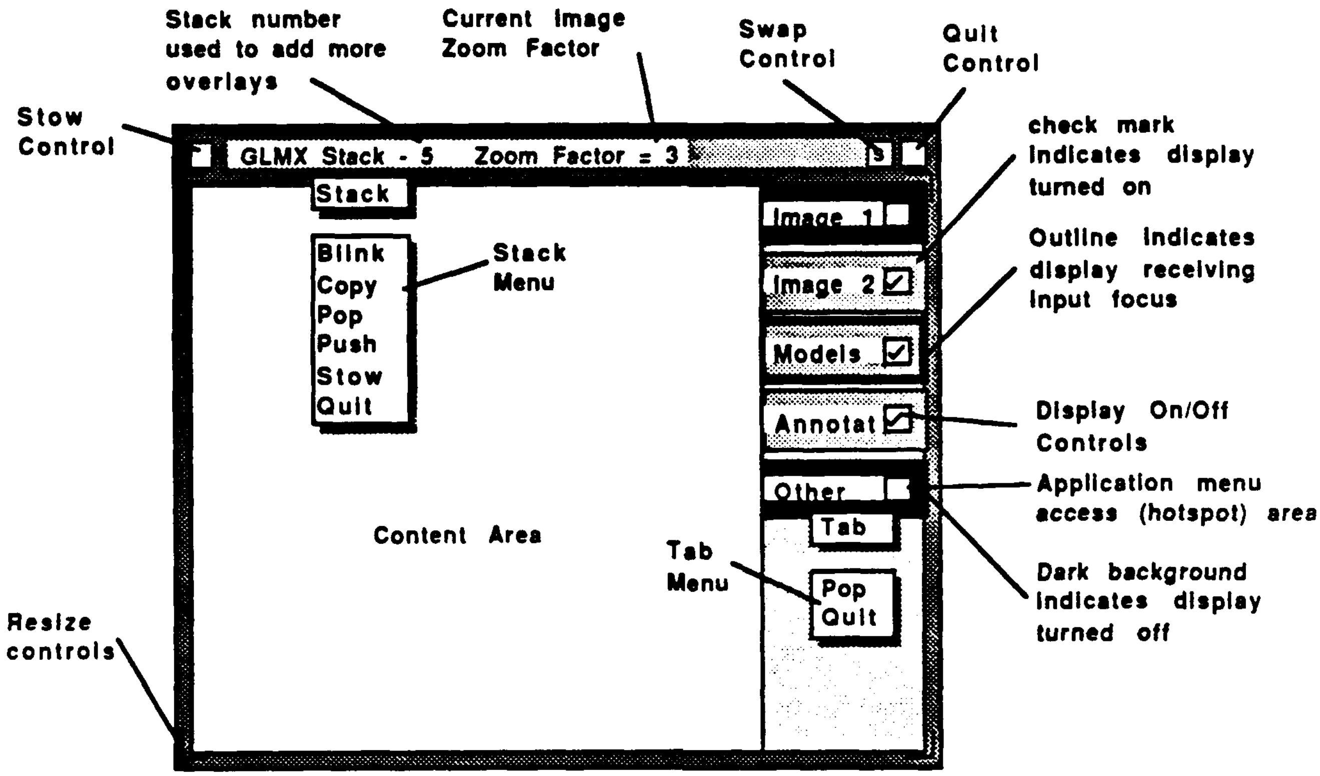


Figure 3-1. GLMX window controls

GLMX Windows have been developed to facilitate the kinds of information fusion capabilities required by the image exploitation task. Some of the key features they provide include:

- Display of imagery and registered graphic overlays
- Zoom, blink, on/off,etc. are handled by the GLMX window manager
- Each overlay is an independent application simplifying development

As depicted in figure 3-1, a GLMX Window can really be considered a stack of information. Some of it can be images, some may be graphic depictions of data base contents, some may be graphic depictions of the results of automatic exploitation algorithms, some could be Image Analyst generated annotation specific to a target

area, or any number of other kinds of useful information. One of the key features of the GLMX Windowing system is that each of these various, sometimes diverse, pieces of information is managed by an independent application whose only job is to present the data to the GLMX Window Manager. All of the application software required to fuse the information is provided as a service of GLMX. This makes it extremely easy to add new sources of information as they become available or necessary during the exploitation task.

Developed in this fashion, GLMX provides an optimal use of limited display space resource. Of course other windows can be simultaneously active at the same time, be they 4Sight or other GLMX Windows, and thus the display space can be allocated among the various tasks being performed by the IA in a fashion best suited to the aspect of exploitation being performed and in a fashion most comfortable for the individual IA.

3.2 Image Analyst Tool Kit

The entire FTCA System is organized and presented to the user in a toolkit fashion. This means the Image Analyst is free to access the available tools in whatever order desired in order to accomplish image exploitation. Some of the tools in the toolboxes are automatic and require only activation by the analyst. Some are interactive and require specific actions by the analyst to perform their function. After logging into the system, the user is presented with a screen which appears as shown in figure 3-2.

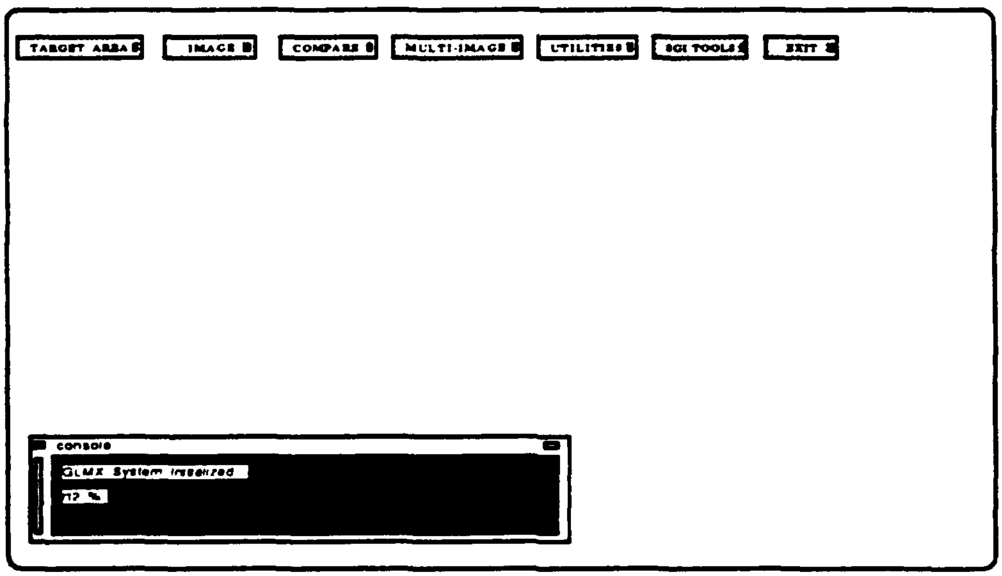


Figure 3-2. Initial FTCA screen layout.

The system's functionality is organized into the 7 toolboxes which appear across the top of the display screen. The console window is a Unix shell window in which informative messages appear during processing. In addition, it can be used to communicate with the Unix operating system and all of the remaining functionality provided by the workstation and Unix.

The functionality within the toolboxes is accessed through a series of menus. When the cursor is over a particular box (e.g. Target Area) and the right mouse button(menu button) is down,the menus associated with that functional area appear. The following list provides an introduction to the menu selections available under each function:

TARGET AREA

Select - select a target area for analysis
Designate -> designate a reference or mission image
Preview ->preview target area
Data Base - place a wireframe overlay of data base on a GLMX stack
Notes - review/create notes for target area to aid interpretation

IMAGE

Display - select an image and display on a GLMX stack
Enhance -> enhance an image in any window
Annotate - place/create target area annotation on a GLMX stack
Statistics - collect statistics over an image window or portion thereof

COMPARE

Run Selected -> select a subset of techniques for comparative analysis processing
Run All- apply all comparative analysis techniques over the selected target area
Review Log - review the results of all comparative analysis techniques
Count -> count an object type over a specified area or predefined count areas

MULTI-IMAGE

Operators -> perform multi-image arithmetic over mission and aligned reference
Align -> perform image alignment

UTILITIES

Get Window Info - overlay text description of any image in a GLMX stack
Lock - Lock the display contents of image areas in multiple GLMX windows
Tracker - track cursor motion in multiple GLMX windows
Remove Image- delete an image file
Mensuration - single or multi-image mensuration

New Camera - define/refine camera model for new image

SGI TOOLS

clock - display a clock
pixel package - pixel zoom, roam, intensity info around cursor
shell - create a new shell window for interfacing with Unix

EXIT

Cancel- do nothing
Clear Memory - clear workstation memory
Exit- log out

Some of the menus are hierarchical and the next level in the hierarchy is selected by sliding the cursor over the arrow symbol which appears. Items containing "->" are hierarchical in nature. The additional layers of menu items will be discussed in the subsequent sections covering each of the major menu item choices.

3.2.1 Target Area Functions

The **Target Area** Function allow the IA to access a number of high level functions related to designating target areas, and the exploitation process related to those target areas. Included are the following functions:

Select
Designate ->
Preview ->
Data Base
Notes

3.2.1.1 Select

The **Select** function allows the IA to select from a list of specified target areas identified by BE designators from a list of target areas maintained within the data base. The Select function initiates a series of prompts that begin with the display of a list of target areas identified by their BE numbers, and terminating with the selection of a target area description file (tadf) for that target area and the particular subdivision that the IA is interested in analyzing. Once the target area description file is selected the system is initialized such that all subsequent functional operations will relate to that particular target area and subdivision identified.

3.2.1.2 Designate

The **Designate** function is a multi-level function that allows the user to designate from a list of images on file a particular image to serve as either the reference or mission image to be analyzed. The designate function may be performed either before or after images are selected for display, and may be changed at any time during the analysis.

3.2.1.3 Preview

The **Preview** function is also a multi-level function that allows the user to preview and become familiar with the target area to be analyzed. The system offers two methods by which a user may become familiar with the target area to be analyzed. The first method allows the user to view an aerial solid model rendering of the target area. The solid model view may be rotated and/or zoomed in or out. The second option on the Preview menu titled "Overlay On Photos" allows the operator to view the target area reference and mission image with the target area 3D data base depicted as a wire frame overlay. Each of the data base structures is identified by a structure number and pointer to that particular structure in the data base. This allows the IA to become familiar with the data base and relate each of the structures within the data base with it's structural identification number. The selection of the Overlay On Photos sub menu option initiates the generation of a GLMX stack containing four tabs (i.e. information layers), a mission image tab labelled MIS, a reference image tab labeled REF, a data base tab and a preview tab. The stack will be initialized such that the mission image will be displayed and the preview and data base functions will be turned on. This will cause the wire frame data base to be overlaid on the mission image (shown in red) and the pointer to structure number one displayed in yellow. By selecting the next option from the preview application menu (which should be popped to the top of the stack), the labeled pointer to the next structure will be displayed. The IA may move forward or backward in the sequence of structure pointer displays by selecting either the next or previous option under the preview menu.

3.2.1.4 Data Base

Selecting the **Data Base** function from the target area menu initiates the data base display application which can be attached to or initiate a new GLMX stack. The data base function computes using the three dimensional target area site model and the associated camera model for the image being displayed, the appropriate perspective transformations and projects the site model as a wire frame overlay on the displayed image. The user may at any time select a new image already on the GLMX stack (by clicking display on/off box) or add a new image at which time the wireframe will be appropriately projected and displayed over the selected image.

3.2.1.5 Notes

Selecting the **Notes** function from the Target Area Menu initiates the display of a notes file in which an IA may review any prior important information about a given target area and may record pertinent information derived from the analysis process. The cursor may be placed anywhere within the notes file for insertion of text from the keyboard. The file supports typical word processing functions such as cut, copy, paste, and search.

3.2.2 Image Functions

The **Image** Function allows the user to perform a number of image related actions as follows:

Display
Enhance ->
Annotate
Statistics

3.2.2.1 Display

When the **Display** Function is requested, the system responds with a list of images covering the selected target area. The user selects an image for display via mouse click and the system responds with a GLMX icon which can be placed over an existing stack or can initiate a new stack by clicking on an open screen area. If placed on an existing stack the system will initiate a new image display application automatically, bring up an application tab, and turn the display on. The image display application has, as part of it's application tab menu, the following options:

Linear stretch
Gray Ramp
New Camera
VLT Stuff
Lead

The Linear Stretch option allows the user to modify the contrast of an image by selecting a window over which the intensities will be examined and a new video lookup table (vlt) defined to stretch the contrast over the full range of intensity levels. The Gray Ramp option resets the vlt to the standard b/w vlt. The New Camera option is used in conjunction with the New Camera Option under the Utility Functions to inform the system that a new camera model is available for use with the displayed image. The VLT Stuff option allows the user to interactively

modify the video lookup table for the displayed image. This supports false coloring, threshold determination and any of a myriad of other possibilities. When selected a new window pops up on the display which shows the current vlt on top and has possible color and b/w mappings on the bottom of the window. The user selects a color by clicking on the bottom half of the vlt window and selects the image intensity(s) over which to apply that color by clicking in the top half of the vlt window. The user can also select a range of colors and interpolate them over a range of intensities by dragging over a range with the left mouse down. The Lead option is a toggle which alternates between "lead" and "follow". If the user wants the display of images on the GLMX stack to track each other then all should be set to be follower (i.e.,, follow whoever is on top).

3.2.2.2 Enhance

The **Enhance** option provides a number of image enhancement techniques as follows:

Histogram Equalization
Robert's grad
2x2 Ortho Grad
2x2 Cross Grad
5 Point Laplacian
9 Point Laplacian
Sobel
Adaptive Sobel
Low Pass Filter
Adaptive Binarizer
Unsharp Masking
Adaptive Unsharp
Dynamic Range Adjustment
Moravec Interest
Image Sharpening
Chien Shadow Oper
Chien Bright Obj

A GLMX window should be open with an image displayed before selecting any kind of enhancement. When selected the user will be presented with a GLMX selection window icon which can be placed over the image area which the user wants enhanced by positioning the window, holding down the left mouse button, and dragging open the window. The system will initiate the enhancement technique, and will initiate a display task which puts up a new tab for the enhanced image when the enhance data is ready for display. The display can be turned on after the

tab appears. The enhanced section of image should appear and be displayed in an aligned fashion with respect to it's parent image.

3.2.2.3 Annotate

When the user selects the **Annotate** Option the system responds with a GLMX icon which can be placed on an existing stack or initiate a new stack. The system initializes the annotation application and adds a tab to the stack for annotation. Any existing annotation for the target area is automatically placed over the displayed image. The user can add or remove annotation as desired as an option under the annotation application menu. A number of forms of annotation are supported including circle, rectangle, polygon, vector, and text. As the user moves from image to image the annotation automatically accompanies each image switch and redraws itself in the correct location on the new image. Thus IA can tie comments, notes, etc regarding the target area to individual images but have them be overlaid on each new mission as it arrives in a registered fashion.

3.2.2.4 Statistics

The **Statistics** Option is used to gather statistical information over a window of image data. When selected the system responds with a GLMX selection window icon. The user can place it, hold down the left mouse button, and drag open the window over the section of image data for which statistical data is desired. The system initiates an application to gather the data and when completed initiates a new tab on the GLMX stack. The system generates a graphic depiction of the image data histogram as an overlay within the GLMX window and also displays the numerical quantities min, max, mean, and variance

3.2.3 Compare Functions

The **Compare** function allows the IA to access a number of specialized automatic comparison and OB detection and counting functions. Included as options are the following:

> **Run Selected ->**
> **Run All**
> **Review Log**
> **Count ->**

3.2.3.1 Run Selected

The **Run Selected** function is hierarchical in nature. The second layer menu which appears when the user slides the cursor over to the arrow is as follows:

Edge Based = on
Neural Net = on
Intensity Based = on
Line Based = on
Shadow Based = on
Run Selected

Included within the FTCA System are multiple techniques for performing comparative analysis between multiple images and a 3D Data Base. The "Run Selected" option allows the user to selectively enable/disable each of the techniques used for target area processing. As the user positions the cursor over the options they will be highlighted. If the right mouse button is released the "on" switch should toggle to "off". This sequence of actions can be repeated for each of the remaining options until the user has the desired set of options enabled. Finally by selecting the "run selected" option. the enabled set of comparison techniques will be performed automatically using the currently designated mission and reference images.

3.2.3.2 Run All

The **Run All** function applies all comparison techniques to the currently designated mission and reference for the selected target area subdivision. This is not dependent upon any settings which may have been made under the **Run Selected** function.

3.2.3.3 Review Log

The **Review Log** function allows the analyst to review the results of automatic comparison processing over the new image. The IA can review the processing results and also have them overlaid as color cues so as to compare and contrast their analysis with the system's.

When selected the user will be presented with a GLMX icon which can be opened. The system will then automatically place the mission, aligned reference, edge, intensity, line, ANS Comparator, shadow verify, and shadow detect results display applications on the stack, each with it's own tab. The applications will be set in an "off" state. The user may selectively turn them "on" by clicking their corresponding tab. When enabled they provide overlay cues on the image over the areas they found to represent change. The color of the overlays will match the color of the tab and identify the comparison technique which is providing the cue. The cues are provided as color overlay masks which allow the IA to see through to the image. Optionally the cues can be changed to oriented bounding rectangles. This option can

be selected by moving to the application menu area of the tab and selecting the "bounding rectangle" option.

3.2.3.4 Count

The **Count** Function allows the IA to detect and count selective OB objects. It is a hierarchical menu item with the next level of the menu containing the following items:

Cars
Tanks
Planes
Count All
Review All

3.2.3.4.1 Count Over Directed Area

The user can direct the system to detect and count OB objects in the car, tank, or plane class. The IA is allowed to select a window over which the detection and counting process will be applied. The system performs detection and counting over the designated image area, and responds by adding an application to the GLMX stack to display the result. The application will be turned on and should place the enclosing rectangle boundary, a mask showing where objects were detected, and an object count as overlays on the screen.

3.2.3.4.2 Count/Review Over Predesignated Areas

The FTCA System also allows for count areas to be pre-designated for the target area. Essentially a number of rectangular areas can be predefined to cover particular ground areas. The object type of interest and normalcy count is stored along with the area bounds. When the **Count All** option is selected, the system processes each of these designated areas. The user can subsequently request to view the results of processing the predesignated areas by selecting the **Review All** option. The system will generate a color-coded overlay for each count area and provide a collective graphic summary for the entire target area from which trend analysis can be determined. Each individual area is enclosed by it's defining rectangle and the object count is displayed in a color indicative of the finding (blue, OB count within normalcy; red, count low; green, count high). In addition, a graphic presentation is provided which shows the collective counts over the last 10 examinations of the count areas. Normalcy is depicted as a horizontal line at the expected count. Deviations will be depicted as deviations from that normalcy line.

3.2.4 Multi-Image Functions

The **Multi-Image** Function allows the IA to access a number of multi-image utility functions. Included are the following:

Operators ->
Align ->

3.2.4.1 Operators

The **Operators** option allows the user to apply any of a number of 2 image operators to a pair of aligned images. By default the current mission and aligned reference are the 2 images utilized. This is a multi-level menu option with the second menu level appearing as follows:

Add
Average
Subtract
Ratio
Threshold
Max
Min
And
Or
Mask
Zero Filter

3.2.4.2 Align

The **Align** function provides a number of image alignment options for the IA. This is a multi-level menu option with the second level menu items as follows:

Pnorm = on
3 Dimensional
Ground Plane

Radiometric correction or alignment of the image intensity data is enabled when the Pnorm toggle is set to "on". Both the 3 dimensional and Ground plane alignment techniques are completely automatic. The system uses the 3D site model for the target area and the camera models of the reference and mission images to effect the generation of an aligned reference image. The current mission and reference image designations determine the source images used for this processing. The reference image data is resampled to produce an aligned version which matches the mission

perspective. Ground plane alignment is similar to global alignment via polynomial transformation. It uses the equation of the ground plane within the site model to effect the alignment. Of the two types, 3 dimensional alignment is preferred because of it's increased accuracy. Ground plane alignment is faster however, and may make sense for some exploitation activities when less precise alignment is required.

3.2.5 Utilities Functions

The Utilities function allows the IA to access a number of specialized utility functions. Included are the following:

Get Window Info
Lock
Tracker
Remove Image
Mensuration
New Camera

3.2.5.1 Get Window Info

This function produces an overlay which provides information on the image being displayed including name, size, display window area, approximate ground sample distance, and number of bits per pixel. Each time a new image is turned on within the GLMX window the overlay updates to provide the information for the image being displayed.

3.2.5.2 Lock

When selected this option can be used to lock the display window contents of two or more GLMX windows to each other. Whenever an area on any image is selected for examination, the other window(s) will automatically display the same area at approximately the same scale. Thus zoom operations are locked between windows as well.

3.2.5.3 Tracker

When selected this option can be used to track the movement of a cursor between the display window contents of two or more GLMX windows. As the cursor is moved on one image all the cursors in the other windows point at the same location. Thus the IA can quickly determine and maintain orientation on multiple images collected from different perspectives.

3.2.5.4 Remove Image

When selected this option allows the user to delete an image file. This option will allow the user to move through directories looking for the image file to be deleted. The search begins at the current target area subdivision image directory.

3.2.5.5 Mensuration

When selected this option allows the user to perform 2 or 3 space mensuration. The application comes up in 2d mode meaning the user can measure lengths on the image and ground using a fixed scale. The user can position the cursor on an object to be measured, hold down the left mouse button, and drag the cursor to the end of the object. As this is being done a line segment appears anchored at the initial cursor placement and following the cursor movement. The length of the line segment in pixels and approximate length in meters on the ground is displayed as an overlay on the image.

Optionally the application performs 3 space mensuration. In 3 space mensuration mode the user is expected to provide 2 space (i.e.,, image) measurements of a common point in the scene.

3.2.5.6 New Camera

The **New Camera** Function allows the generation of a camera model for a new mission image for which a 3D site model data base and existing reference image are available. The process by which the camera model is built for new mission image requires the selection of conjugate points between the site model and the mission image for which the new camera model is being built. In order to build the new camera model approximately eight to ten conjugate points between the mission image and data base must be selected. The existence of a site model which can be projected as a wire frame over an existing reference image helps in the process of picking reference data points. Following the selection of approximately eight to ten conjugate points, the user can initiate the building of a camera model for the new mission image. This camera model can be subsequently refined until a highly accurate alignment of the target area data base to the new mission image is achieved.

3.2.6 SGI Tools

The **SGI Tools** function allows the IA to access a number of window-oriented tools provided with the IRIS 4D workstations. The options include:

clock
pixel package

shell

3.2.6.1 Clock

When selected this option presents the user with a clock icon. The analyst can use the clock to measure his efficiency or alarm when it's time to go home.

3.2.6.2 Pixel Package

When selected this option opens 3 windows at the bottom of the display screen. The first provides a zoomed version of the area around the cursor location. This window comes up in color map mode, but if the center mouse button is pushed down the user should see a b/w intensity version of the image. Intensity at the cursor in both the front and back buffers are displayed as well as screen coordinates. The center window contains the video lookup table which can be modified by depressing the right mouse button and selecting the appropriate action. The right window contains 3 slide bars which can be used to edit a VLT entry.

3.2.6.3 Shell

When selected this option allows the user to generate a new shell window for communication with the Unix operating system

3.2.7 Exit Functions

The Exit Function allows the IA to exit FTCA processing as you might expect. The options under the Exit menu include:

Cancel
Clear Memory
Exit

3.2.7.1 Cancel

If selected this option cancels the exit request.

3.2.7.2 Clear Memory

If selected this option clears the display and RAM available to the CPU. This can be used to clear the memory after classified processing. Before selecting this option all GLMX Windows should be closed.

3.2.7.3 Exit

When selected this option logs the IA out of the FTCA System. The screen will temporarily go blank and will be followed by the account presentation icons waiting for the next user to log in.

3.3 Model Supported Exploitation

The 3D site models facilitate the comparative analysis over target areas for both the Image Analyst and the automatic comparative analysis software processes. Each of these will be described in the subsequent sections.

3.3.1 IA Support

In the case of the IA support, the site models provide an information cueing basis for rapid familiarization and visual inspection. The ability to graphically overlay the site model on new mission image as it is collected provides an extremely low risk and rapid technique for allowing the IA's focus-of-attention to be immediately directed to the objects of interest within the target area.

In addition, the site model provides a common, extensible information knowledge base for storing all relevant data about the target area. There was insufficient time to explore this fully during the contractual effort. The system provides the basis for exploring the concept by storing target area data in the annotation, notes, and site model data base files. These could be integrated in a better fashion and provide a much better tool for the IA. Information currently distributed over these three files could be cataloged in a better fashion and selectively recalled on an object-by-object basis as needed by the IA.

The other very important utility which the site model provides for the IA is a fixed coordinate system for presenting information in a fused fashion. Given multiple images covering the target area, the site model allows all images and all information associated with each image to be tied together via the site model. The camera model defined for each image, provides the sensor position and attitude data to allow transformation from 3 space in the site model to 2 space image coordinates. This coordinate transformation capability is subsequently used by the GLMX Windowing System to make the examination of image data and processing results from automatic comparison techniques very easy for the IA.

3.3.2 Automated Comparative Analysis Support

In addition, the site model serves to provide a basis for automating the comparative analysis task. We explored several approaches to this which will be described in

section 4. We believe these represent an initial sample of the class of advanced techniques which can be brought to bear given the existence of the site model.

The geometry information allows for prediction and directed search for explicit target signatures in the mission image to confirm the continued existence of objects from the site model. The geometry information when coupled with the sun position at image collect time allows for the prediction and directed search for shadows from known site model objects. The geometry information provided by the site model when coupled with the image camera models allow precise image intensity data alignment.

The above are some of the aspects of the site model which FTCA utilized for automating the comparative analysis process. There are many others which could be developed depending upon the model completeness. For example, additional techniques relying on surface material reflectivity techniques could be developed. Other examples would be directed search for specific attributes of site model objects which make them unique (e.g., window, door, ventilator, etc. combinations). Beyond the confirmation of objects which should be present is the detection and cueing of changes. Changes result because of change to existing objects or addition of new objects.

The FTCA system detects and cues areas of change but performs no reasoning about the nature of the changes. Furthermore FTCA provides multiple techniques for detecting change between the mission and aligned reference images and/or between the site model and mission image. The outputs of each of these individual techniques could be combined into a more concise presentation for the analyst. Changes cues could be prioritized depending upon the number of techniques which agree that a change is present in an area and the confidence each has that the area is a change. The other area where considerable development is required is in the area of reasoning (i.e. explaining the nature of the change). This is an extremely difficult and ongoing research area. It is not clear that technology is mature enough to provide any significant help to the analyst in this area, so this would be a high risk development area to pursue.

4.0 ALGORITHM DETAILS

The following sections provide additional implementation detail on the key algorithms delivered within the FTCA System. Included are local operators applied to enhance the source imagery in support of various comparative analysis techniques and/or for improved analyst interpretability. In addition, the various comparative analysis techniques are described. A description of the procedure for new image camera model definition is provided. The image alignment process which utilizes the 3D site model is described. Finally the provision within FTCA for Order of Battle (OB) Object Detection and Counting is also described.

4.1 Local Image Processing Operators

Local operators take a small area of image into consideration at each time. Usually, an n x n pixel square is used to define the subimage for the processing. Local operators are simple and can be easily implemented. In many cases, they are very effective.

Several local operators have been developed and implemented within the FTCA system. Some of them are utilized to do:

1) Edge preprocessing and detection;
2) Image enhancement; and
3) Bright-object and dark-object/shadow detection.

Additional local operators which use binary images as inputs were also implemented. When a binary image consisting of feature and non-feature pixels identified by other operators is processed, local operators are used to do:

1) Distance transform which assigns a value to all pixels corresponding to the distance away from the nearest feature pixel;

2) Maximum value expansion which extends a local maximum computed from a binary distance transform or other type of measure such as vote-count in the line detection(explained below), to all pixels within a connected area or contour; and

3) Feature labelling which labels all pixels in the feature with an identification number.

These local operators will be described and discussed in the following sections.

4.1.1 Edge Preprocessing and Detection

There are two major processing steps involved in finding edge pixels in a gray-level image. The first step is to preprocess the image so that at each pixel of the processed image, the strength and the direction of a possible edge are produced. In the second step, an edge point is detected based on the information obtained in the first step.

4.1.1.1 Edge Preprocessing

After considering many different edge operators, a moment-based edge operator [1] was selected and implemented for the edge preprocessing. It uses an ideal step-edge model to derive a set of equations in which every idealized edge parameter is expressed as a function of two-dimensional spatial moments over a circular region. These moments are estimated from the image data over a fixed window. Thus the strength of the step-edge can be calculated from estimated moments. The matching between image data and an ideal model is indirectly accomplished through the matching between model-derived moments and estimated moments.

The 2-D spatial moment of an image I(x,y) of order (p,q) is given by

$$M(p,q)=\iint x^p y^q \, I(x,y)\, dx\, dy. \qquad (1)$$

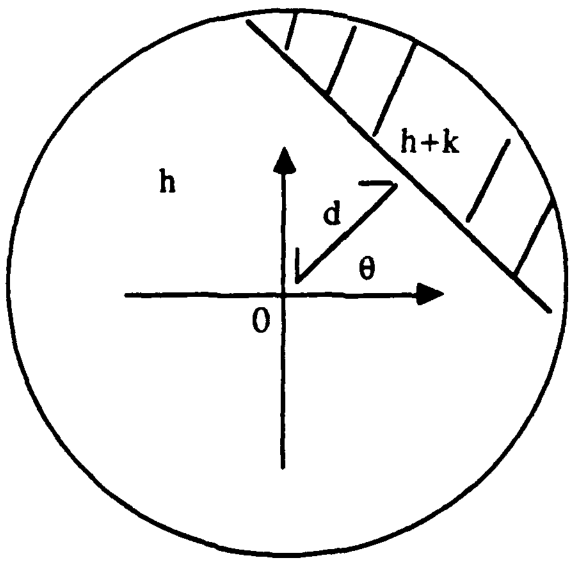


Figure 4-1. Model of an ideal step-edge.

If the model of an ideal step-edge as shown in Figure 4-1 is assumed, the results for the edge parameters in terms of the moments up to order two (M(0,0), M(0,1), M(1,0), M(1,1), M(2,0), M(0,2)) are as given below and provided in [1,2] :

$$k=(3Ma)/(2\sqrt{((1-d^2)^3)}), \tag{2}$$

$$\theta = \tan^{-1}(M(0,1) / M(1,0)), \tag{3}$$

$$d=Mb/Ma^3 - Mc/Ma; \tag{4}$$

$$Ma=\sqrt{(M^2(0,1) + M^2(1,0))}; \tag{5}$$

$$Mb=4[M^2(1,0)\, M(2,0) + 2M(0,1)\, M(1,0)\, M(1,1) + M^2(0,1)\, M(0,2)]/3; \text{ and} \tag{6}$$

$$Mc=M(0,0)/3; \tag{7}$$

where k is the edge magnitude, θ is the edge orientation, and d is the distance between the center and the edge.

Equations 2)-(7) indicate that the problem of finding edge parameters is solved if 2-D moments can be estimated from the image data. The estimation of moments is achieved by defining a circular region on an n x n pixel window and assuming a uniform intensity value over the area occupied by the pixel.

As shown in Figure 4-2(b), a circular region is defined on a 3x3 pixel window with pixel intensity I(i,j), $-1 \le i,j \le 1$ (refer to Figure 4-2(a)). Now, from (1), we have

$$M(p,q)= \sum_{i=-1}^{1} \sum_{j=-1}^{1} I(i,j) \iint_{A(ij)} x^p y^q \, dx\, dy. \tag{8}$$

Let $$W(i,j,p,q) = \iint_{A(ij)} x^p y^q \, dx\, dy. \tag{9}$$

then (8) becomes

$$M(p,q) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} W(i,j,p,q)I(i,j). \tag{10}$$

Equation (10) indicates that moments are calculated by correlating spatially-determined moment weights with the image data. Thus once moment weights are determined, the calculation of moments is the same as applying a local operator to the image within a small window.

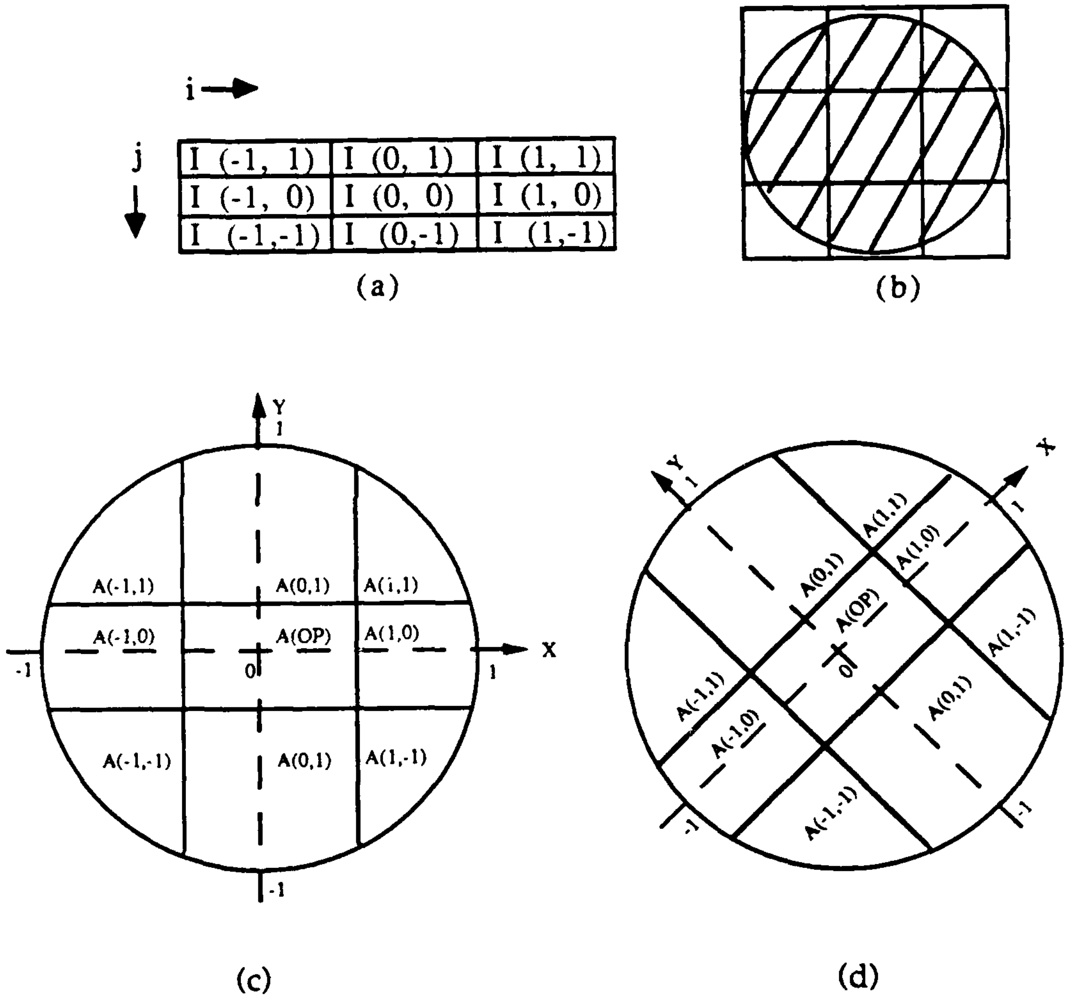


Figure 4-2 (a) A 3x3 pixel window with intensity value I(i,j), $-1 \le i,j \le 1$;
(b) A circular region defined on a 3x3 pixel window;
(c) A circular region with unit radius divided into 9 uniform intensity areas by a 3x3 pixel window;
(d) Same as (c), except x,y-axis rotated 45 degree

From (9), moment weights are the results of carrying out double integration over an area defined by both the circular region with unit radius, and the pixel window. As an example, moment-weight masks for moments M(1,0) and M(0,1) within a 3x3 window are

1) xy-axis in parallel with the sides of window (Figure 4-2(c)):

M(1,0)

-.137375	0.0	.137375
-.283950	0.0	.283950
-.137375	0.0	.137375

M(0,1)

.137375	.283950	.137375
0.0	0.0	0.0
-.137375	-.283950	-.137375

2) xy-axis rotated 45 degree (Figure 4-2(d)):

M(1,0)

.194278	.200783	0.0
.200783	0.0	-.200783
0.0	-.200783	-.194278

M(0,1)

0.0	.200783	.194278
-.200783	0.0	.200783
-.194278	-.200783	0.0

It has been shown above that 2-D spatial moments are utilized to calculate ideal step-edge parameters, and moments are directly estimated from pixel values within a predefined window. As shown in (2) and (3), moments M(1,0) and M(0,1) are sufficient for the calculation of edge orientation. However all of moments up to order two are required for the calculation of edge magnitude. The computational complexity is much higher than other local operators such as Sobel, and Prewitt edge operators.

Now, back to (2) and let

$$k'=k\sqrt{((1-d^2)^3)}. \qquad (12)$$

Then,

$$k'=3\sqrt{(M^2(1,0)+M^2(0,1))}/2. \qquad (13)$$

If the step edge passes through the center of the window, i.e., d=0, k' equals k. But, if the step edge is away from the center, i.e., d is not zero, k' becomes less than k And k' decreases as the edge moves further away from the center. This indicates that the value of k' can be utilized to decide on edge pixels within a local area. In this case, an edge pixel is indicated by the local maximum of k' along the ridge.

Since the step edge does not always pass through the center of the window, the inaccuracy introduced by using k' as the edge magnitude may cause erroneous detection of edge pixels. However, as pointed out in [2], the magnitude error due to the use of k' is not too serious and it is quite similar to that generated from the Sobel edge operator. The advantage of course is that the computation complexity is considerably reduced. This is because the same M(1,0) and M(0,1) for the calculation of edge orientation are required for the calculation of k'.

If 3x3 moment-weight masks for M(1,0) and M(0,1) shown above are normalized with the minimum nonzero weight, the new masks become:

Case 1) Normalization factor: .137375

M(1,0)			M(0,1)		
-1.0	0.0	1.0	1.0	2.0669	1.0
-2.0669	0.0	2.0669	0.0	0.0	0.0
-1.0	0.0	1.0	-1.0	-2.0669	-1.0

Case 2) Normalization factor: .194278

M(1,0)			M(0,1)		
1.0	1.0335	0.0	0.0	1.0335	1.0
1.0335	0.0	-1.0335	-1.0335	0.0	1.0335
0.0	-1.0335	-1.0	-1.0	-1.0335	0.0

Case 1) is very close to the Sobel edge operator. And case 2) is quite similar to the rotated Prewitt edge operator. It is interesting to notice that the ratio between the maximum weights of both cases is

$$.283950/.200783 = 1.4142 \approx \sqrt{2}.$$

This indicates that the edge magnitude calculated from the Sobel operator approximately equals $\sqrt{2}$ times that calculated from a 45 degree rotated Prewitt operator.

Sobel, Prewitt, and Kirsch edge operators are generally considered as the compass gradient edge operators [3,4]. The factor of $\sqrt{2}$ relationship between the Sobel and Prewitt has also been previously suggested. Based on the results of moment-based

edge operator as defined above, the $\sqrt{2}$ relationship is rather between the Sobel and 45 degree rotated Prewitt.

4.1.1.2 Edge Detection

The moment-based edge operator correlates a region of the image to a set of 2-D spatial moments which are then matched to a set of idealized edge parameters. This method is considered as a regional edge detection method. Since the calculation results relate to the matching between the image data and an ideal model, a confidence measure that indicates how well the image data matched the model is obtainable. This measure is usually utilized to detect the presence of edge pixels.

However, in the derivation shown above, a simplified edge magnitude calculation which converts a regional edge operator to a local edge operator is suggested. So, the edge detection method will be modified to reflect such change. The method takes the advantage of decreased magnitude response as the edge moved away from the center of the pixel window. Clearly, as indicated in [13], this implies a combination of thinning and thresholding process for the edge detection.

4.1.1.2.1 Thinning

It is well known that the local edge operator produces thick edge magnitude contours. In order to extract edge contours, a thinning process is needed. Normally, the presence of an edge at a pixel is decided by comparing the edge magnitude with its two neighbors in a direction normal to the direction of this edge. In this case, both edge magnitude and edge orientation are required.

A different thinning method is developed in this program. This method does not check the edge orientation for the selection of its two neighbors. Instead, it compares the edge magnitude of the center pixel with that of its eight nearest neighbors. The eight neighbors are grouped into four pairs as shown in Figure 4-3. The magnitude of the center pixel is then compared to the maximum magnitude in each pair. If it is greater than or equal to two or more pairs, it is detected as an edge pixel.

A	B	C
D	x	D
C	B	A

Four pairs: A, B, C, and D.

Figure 4-3. Eight neighbors grouped into four pairs.

In fact, there are four possible cases to be considered for the detection of edge pixels as follows:

The magnitude of the center pixel is greater than or equal to the maximum magnitude of:

1) one or more pairs,
2) two or more pairs,
3) three or more pairs, or
4) all pairs.

Case 1) keeps too many non-edge pixels. Case 2) keeps some extra none-edge pixels and drops a few edge pixels. But, the thinned image shows the best connection of edge contours. Case 3) drops too many edge pixels. Case 4) detects local maximum only.

4.1.1.2.2 Thresholding

The edge operator transforms an intensity image into an edge magnitude image. After thinning, all remaining pixels are treated as edge pixels. But, many unwanted edge pixels are still visible in the thinned image. Thresholding is a simple way to eliminate most of those unwanted edge pixels, since edge pixels with low edge magnitude are mainly detected from isolated homogeneous region or noisy areas. On the other hand, those with high edge magnitude are mainly boundary pixels between homogeneous regions. In many cases, they represent feature boundaries or feature edges. Normally, there is no clear single cut between high and low edge magnitude. However, a properly selected threshold can significantly clean up non-feature edge pixels.

After studying and experimenting with several thresholding techniques found in the literature [5-8], a technique presented by Ostu [8] and modified by the others [6,7] was selected and implemented. This technique determines a threshold value from the image histogram which maximizes the interclass variance between dark and bright regions. In [6] the interclass variance between these regions was formulated, and a simple formula was analytically derived from which a fast search scheme was subsequently developed. Interestingly, in [7], an iterative method to find thresholds based on Ridler and Calvard's work [9] was also proposed. Thus multiple investigations all tried to solve an identical problem, and they utilized the same technique to determine multiple thresholds for image segmentation.

The iterative method found in [7] was implemented for FTCA. The iterative procedure for finding a single threshold is defined as follows:

Assume the minimum and maximum value in the histogram is 0 and N, respectively, and m(i,j) is the mean value of all image pixels with value between i and j. If in the histogram, p(n) is the number of pixels with value n, then

$$m(i,j) = \sum_{n=i}^{j} n\,p(n) \Big/ \sum_{n=i}^{j} p(n). \qquad (14)$$

The iterative procedure is:

1) Choose an initial guess of the threshold as Ta;
2) Calculate m(0,Ta), and m(1+Ta,N);
3) Calculate new threshold Tb using

$$Tb=[m(0,Ta)+m(1+Ta,N)]/2; \qquad (15)$$

4) Check if Ta=Tb is true.
   True: Stop and output Tb as the threshold;
   False: Replace Ta by Tb, and repeat steps 2-4).

A similar iterative procedure can be extended to find multiple thresholds. In this case, the new ith threshold is determined by:

$$Tb(i)=[m(1+Ta(i-1),Ta(i))+m(1+Ta(i),Ta(i+1))]/2 \qquad (16)$$

The process stops when all new and old thresholds are unchanged.

4.1.1.3 Edge Enhancement and Detection Summary

The procedure for the edge enhancement and detection is summarized as follows:

1) Apply a 3x3 moment-based edge operator, assign the edge magnitude k' to the center pixel of the window, where

$$k'=3\sqrt{(M^2(1,0) + M^2(0,1))}\,/2; \text{ and}$$

$$M(1,0) =[aI(1,1) + bI(1,0) + aI(1,-1)] -[aI(-1,1) + bI(-1,0) + aI(-1,-1)];$$

$$M(0,1) = [aI(-1,1) + bI(0,1) + aI(1,1)] -[aI(-1,-1) + bI(0,-1) + aI(1,-1)];$$

with a=.137375 and b=.283950.
Figure 4-4a depicts an edge preprocessed image.

2) Thin the edge image using the technique described in 4.1.1.2.1. Figure 4-4b shows a thinned edge image.
3) Produce an edge magnitude histogram from the thinned image.
4) Find a single threshold from the histogram using the method described in 4.1.1.2.2.
5) Threshold the thinned image and produce a binary edge image. Figure 4-4c shows a thinned and thresholded image.

Figure 4-4a. An edge preprocessed image.

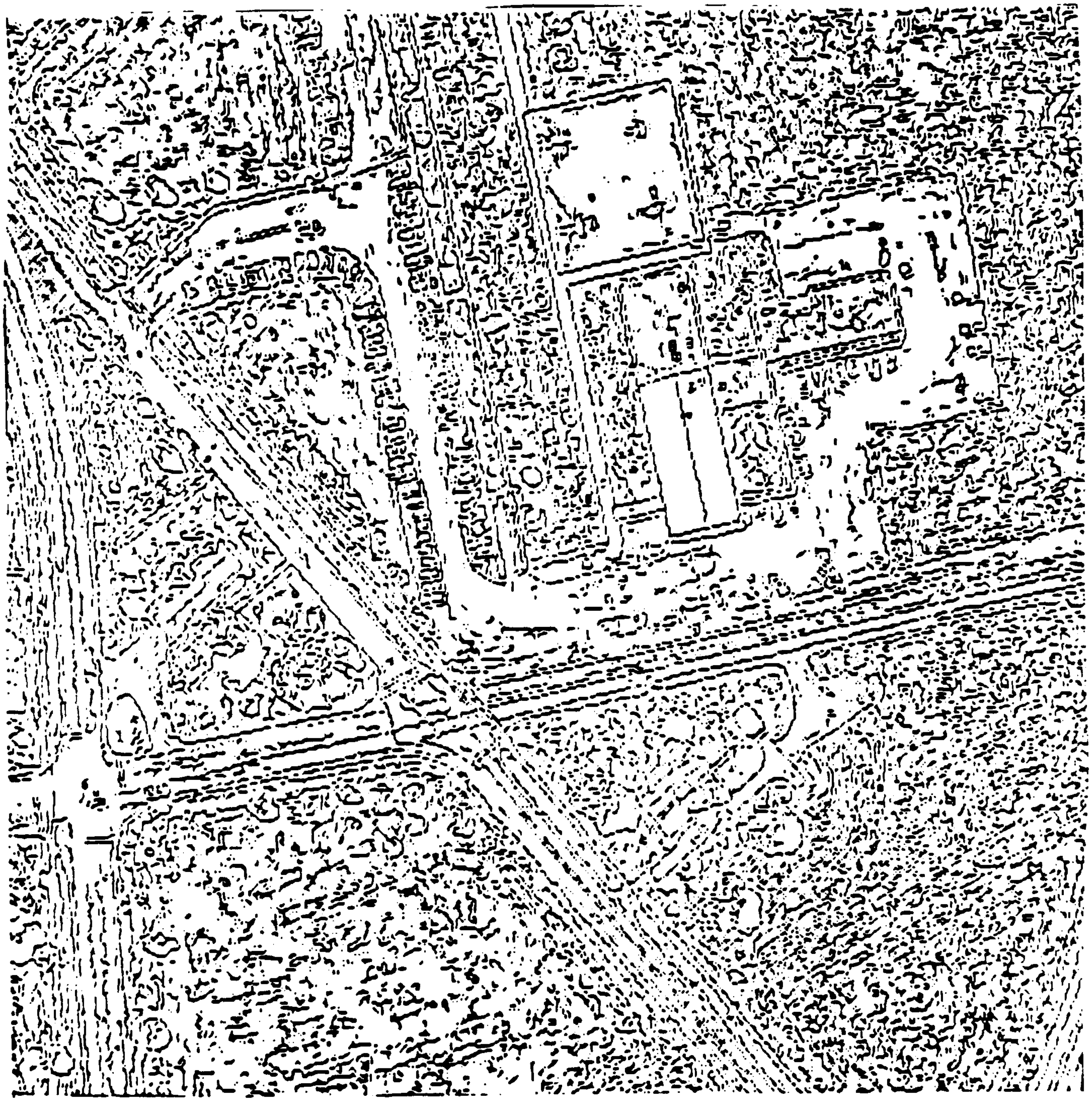


Figure 4-4b. A thinned edge image.

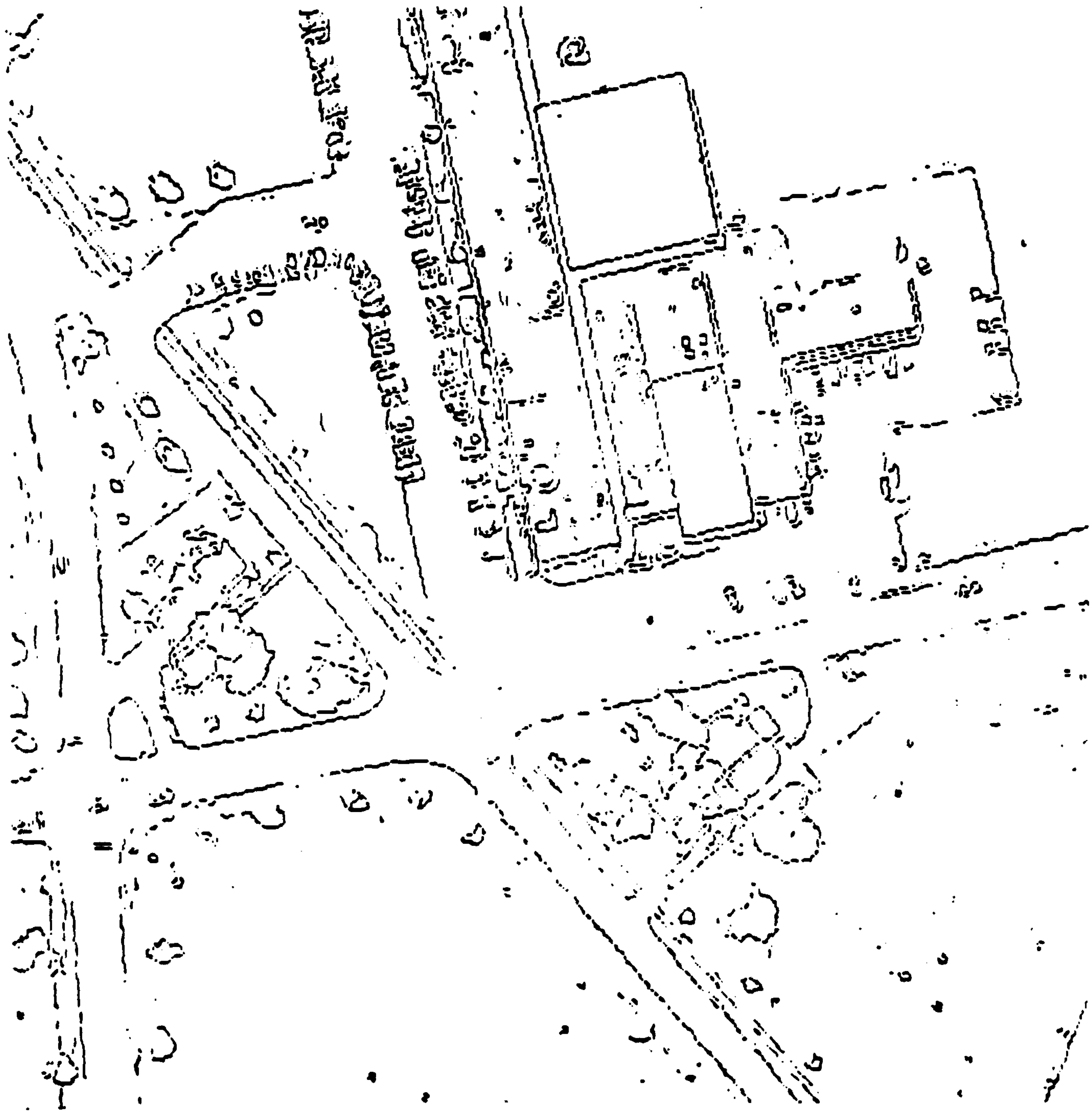


Figure 4-4c. A thinned and thresholded image.

4.1.2 Image Enhancement

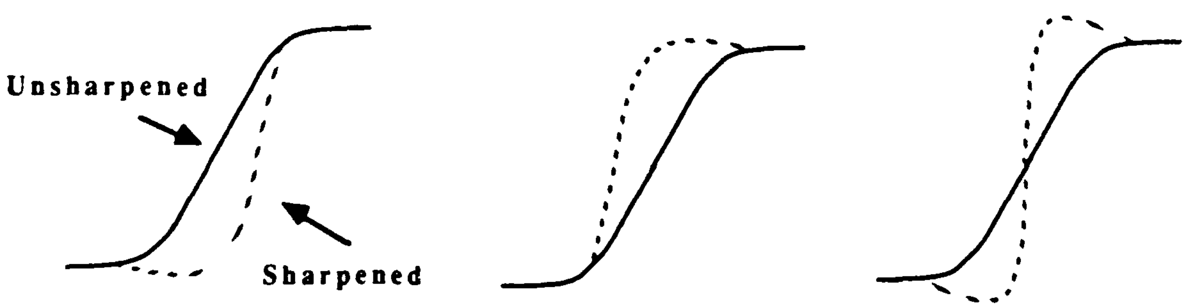
Image enhancement techniques are usually employed to improve image quality for viewing, or feature extraction and recognition. There are many image enhancement techniques available for different purposes. In [10], they are classified into four categories: spatial smoothing, gray-level rescaling, edge enhancement, and frequency-domain filtering.

As part of the FTCA System, edge enhancement and image sharpening operators were requested. These image enhancement techniques were requested to deblur object edges in an image for the image analyst. A simple way to achieve this is to enhance the contrast at edges, i.e., to make bright pixels brighter and dark pixels darker at the edge. This can be achieved by utilizing spatial domain edge enhancement techniques, or frequency-domain high-pass filtering techniques. The frequency-domain technique usually requires a 2-D FFT process to manipulate image data, and a set of filtering parameters to specify desired frequency response. The computation of image data and the selection of filter specifications can be very time consuming. Spatial-domain techniques were implemented as a series of local operators within FTCA.

4.1.2.1 Spatial-Domain Technique for Image Sharpening

A spatial-domain technique for image sharpening was developed for FTCA. It is designed to make bright edge pixels brighter and/or dark edge pixels darker. The amount of enhancement is controlled by two independent parameters; one is for the brightness and the other is for the darkness.

Figure 4-5 illustrates the image sharpening concept. In order to implement the concept, a quantitative measure indicating the brightness or darkness must be established.



(a) Dark darker (b) Bright brighter (c) Both

Figure 4-5. Image sharpening concept at the edge.

Consider a 3x3 window, and let Imax and Imin be the maximum and minimum pixel intensity in the window, respectively, then calculate the difference between the center pixel intensity, and the maximum and minimum pixel intensity:

$$DImax = Imax - I; \quad (17)$$
$$DImin = I - Imin; \quad (18)$$

where I is the center pixel intensity.

In this case, two local extremes are utilized as the references for the brightness and darkness. So, (17) indicates how dark the center pixel is when it is compared with the brightest pixel in the window. Therefore, DImax can be chosen as the measure for the darkness. Similarly, DImin is the measure for the brightness.

If DImax > DImin, or I < (Imax+Imin)/2, the center pixel is classified as a dark pixel, or, vice versa for a bright pixel. Notice that if there is a strong edge present in the window, the difference (Imax-Imin) can be significant. This, in turn, produces a higher difference value in DImax or DImin. So, the brightness or darkness measure is also a good edge strength indicator. This property will be used to assist the processing of car detection and counting described in later sections. Once the darkness and brightness are established as in (17) and (18) respectively, they can be employed to achieve image sharpening results depicted in Figure 4-6 according to rules (a), (b), and (c) as follows:

Case (a) If I is not the brightest, then make it darker, i.e.,

$$I' = I - U \times DImax; \quad (19)$$

Case (b) If I is not the darkest, then make it brighter, i.e.,

$$I' = I - V \times DImin; \quad (20)$$

Case (c) If (U x DImax) > (V x DImin), then make I darker, Otherwise, make it brighter, i.e.,

$$I' = I + V \times DImin - U \times DImax. \quad (21)$$

where I' is the enhanced pixel intensity; U and V are parameters for controlling the amount of image enhancement. Clearly, this technique requires the specification of U and V.

By applying this technique to our test images, good sharpened images based on visual inspection are produced from the following two sets of U and V:

1) U=1 and V=0, or

$$I'=2I-Imax. \tag{22}$$

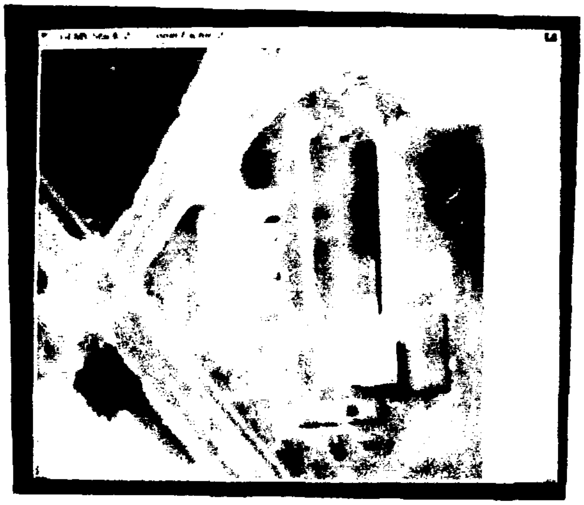
In this case, all non-brightest pixels are darkened.

or 2) U=(255-I)/50 and V=I/50, or

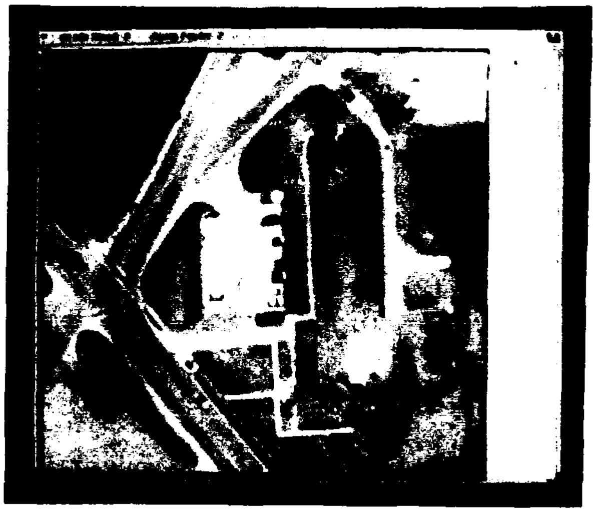
$$I'=I+(DImin \times I)/50-DImax\ (255-I)/50. \tag{23}$$

In this case, global maximum (255) and minimum (0) are utilized to determine the amount of enhancement. For example, locally (within a 3x3 window), the center pixel is decided as a bright pixel. But, it can be globally a bright or dark pixel. For the first case, the edge enhancement operator will make it brighter. But, for the second case, the operator will not enhance it as much as in the first case.

Both combinations of U and V are implemented in this program. If equation (22) is applied to the image shown in Figure 4-6a, the enhanced image resulting is depicted in Figure 4-6b.



a. Source



b. Sharpened

Figure 4-6. Image sharpening.

4.1.3 Bright-Object and Dark-Object/Shadow Detection

A visually noticeable object usually has strong intensity changes near object boundaries. When an object is considered as a bright object or a dark object, the intensity of the object shall be quite bright or dark, respectively. These two properties are utilized to detect bright or dark objects. Shadows have the same properties as dark objects. Thus shadows are included in the dark object category.

(17) and (18) in 4.1.2.1 specify the calculation of darkness and brightness at the edge, respectively. If they are combined with the information on pixel intensity, a good bright-object or dark-object indicator is produced. The bright or dark object pixel is then decided based on the strength of the indicator. This is done as follows:

1) Calculate: (using a 5x5 window)

   a) $I1 = DImax/(1+I-Gmin);$ (24)

      where I1 is the dark-object indicator.

   b) $I2 = DImin/(1+Gmax-I);$ (25)

      where I2 is the bright-object indicator.

2) If I1 is greater than or equal to one, a dark-object pixel is detected; if I2 is greater than or equal to one, a bright-object pixel is detected.

In (24) and (25), the Gmin and Gmax are the global minimum and maximum, respectively. For an 8-bit gray-level image, they are usually set to 0 and 255, respectively. In this case, I1 and I2 become:

$$I1 = DImax/(1+I); \text{ and} \quad (26)$$
$$I2 = DImin/(256-I). \quad (27)$$

As shown above, the information on intensity changes and intensity itself are combined in a way that the darkness and brightness are divided by a factor relating to how bright or dark the center pixel is globally, respectively. Notice that DImax and DImin are obtained from a 5x5 search window. A wider window is favorable for shadow and small Order of Battle (OB) object detection (described latter). Figure 4-7a shows a bright object detected image; and Figure 4-7b shows a dark object/shadow detected image near the building 240 area of Griffiss.

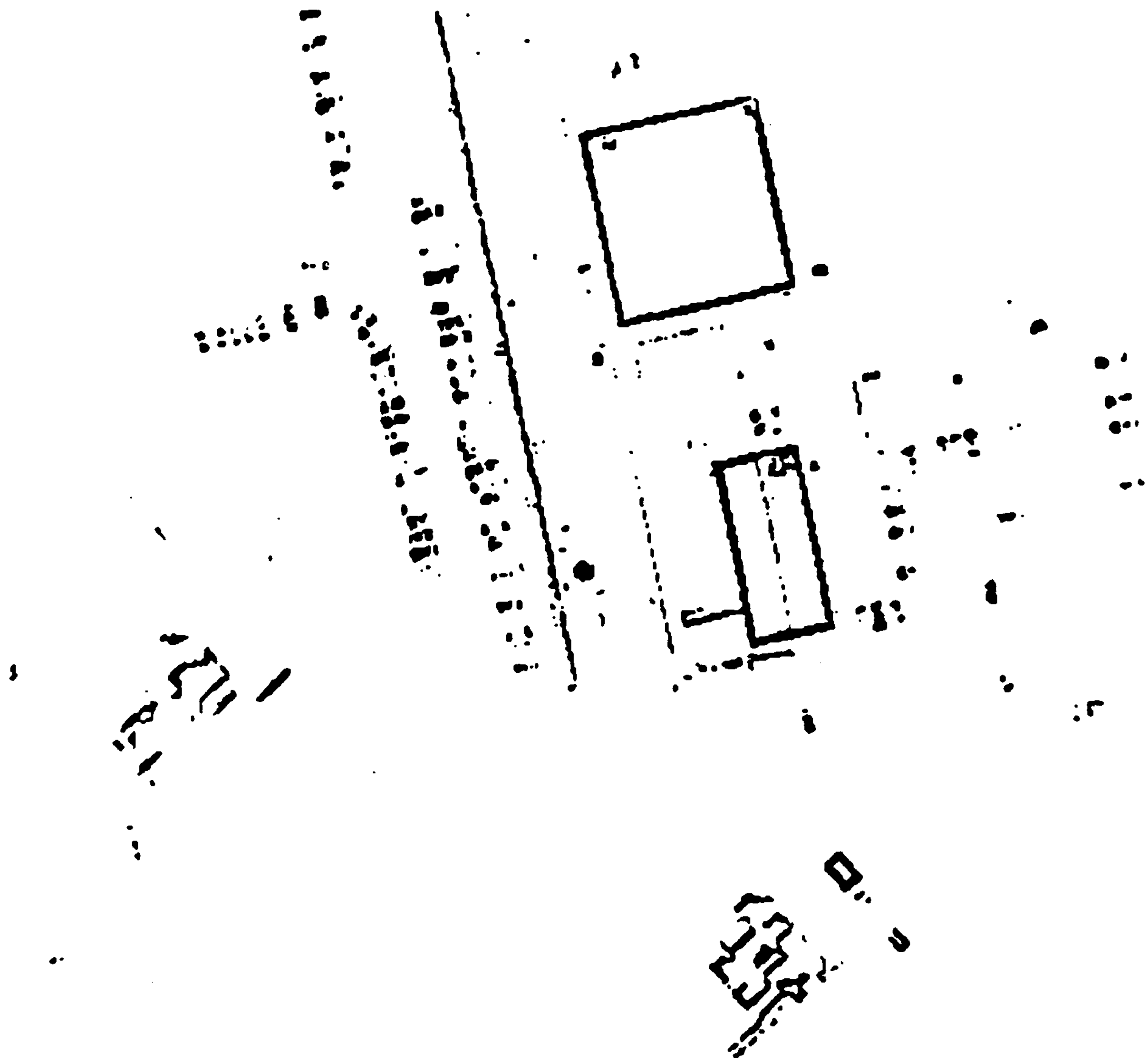


Figure 4-7a. Bright object detection image.

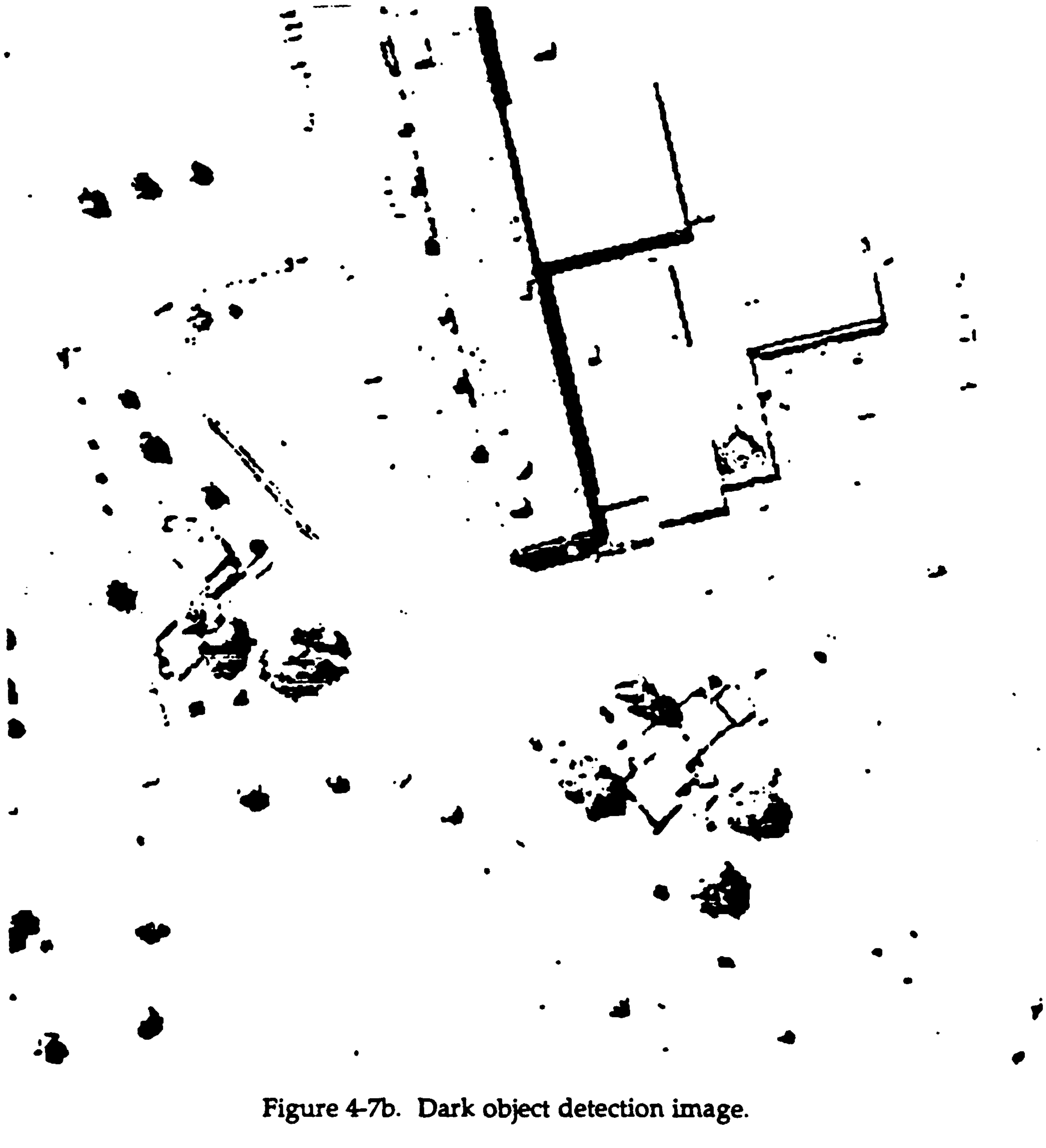


Figure 4-7b. Dark object detection image.

4.1.4 Distance Transform

A decision operator such as edge detection, and bright or dark object detection converts a gray-level image to a binary image in which a 'one' indicates a pixel of interest (be it edge or object), and a 'zero' indicates a background pixel. On the other hand, a binary distance transform (DT) converts a binary image back to a multi-level image in which all pixels in the binary image are assigned a value in the multi-level image corresponding to the distance to the nearest 'one' pixel. As indicated in [11], examples of DT applications are merging and segmentation, skeletonizing, clustering, and matching.

There are many different families of distance transformations. The most popular one is the 8-neighbor DT (city-block/chessboard DT) which was implemented. The computation of the DT can be either parallel (global) or sequential (local). Since the implementation of a local operator is simple and fast, the sequential algorithm described in [11] is adopted.

The sequential algorithm requires two passes over the image. The first pass is the forward pass which processes image from left to right and from top to bottom; and the second pass is the backward pass which processes image from right to left and from bottom to top. The procedure is described as follows:

1) Convert a binary image into a zero/infinite image, i.e., assign "one" in the binary image to "zero" in the new image; and assign "zero" in the binary image to the largest number allowed in the new image. Let the new image be a M rows by N columns image, and the value at the (i,j)th position be I(i,j), where $1 < i < M$ and $1 < j < N$.

2) Process forward: (using a 3x3 window)

    For i=2 to M do
        a) For j=2 to N-1 do
            $$I(i,j)=\mathrm{Min}[I(i-1,j-1)+1,\ I(i-1,j)+1,\ I(i-1,j+1)+1,\ I(i,j-1)+1,\ I(i,j)] \quad (28)$$

    At the end of a), do b)
        b) For j=N-1 to 1 do
            $$I(i,j)=\mathrm{Min}[I(i,j+1)+1,\ I(i,j)]; \quad (29)$$

where Min[a,b,..,i] is an operation to find the minimum among a,b,..,i.

3) Process backward:

For i=M-1 to 1 do
a) For j=N-1 to 2 do

$$I(i,j)=\mathrm{Min}[I(i+1,j+1)+1,\ I(i+1,j)+1,\ I(i+1,j-1)+1,\ I(i,j+1)+1,\ I(i,j)] \quad (30)$$

At the end of a), do b)
b) For j=2 to N do

$$I(i,j)=\mathrm{Min}[I(i,j-1)+1,\ I(i,j)]. \quad (31)$$

4.1.5 Maximum Value Expansion

One of the DT applications is to do image merging and segmentation. The maximum value (MV) expansion technique is designed to convert distance transformed or other related images to a form suitable for image merging and segmentation. For example, after edge detection, object boundaries are detected. If one wants to count the total number of pixels occupied by the object, it is not easy to get a correct count from the edge image directly. But, if pixels inside object boundaries are filled in and all object pixels are labelled, the pixel counting becomes easy and reliable. The maximum value expansion technique can be employed to fill pixels inside the object. In the next section, a labelling technique will be described. Both techniques are directly derived from the DT technique.

What does the maximum value expansion technique do? It simply extends the maximum value within a bounded common region to every pixel in the region. In the example mentioned above, the procedure to fill in an object is as follows:

1) Do DT to the edge detected image;

2) Apply the maximum value expansion technique to the DT image. Now, all pixels inside the object are assigned a common distance number which corresponds to the maximum size of the object, and all pixels between different objects get a different common distance number which is significantly larger than that associated with objects. For example, if the largest distance number inside the object is M, then M will be assigned to all pixels originally with a number less than M in the same region.

3) Threshold the MV image and keep all pixels with a value less than the threshold.

The algorithm for the maximum value expansion technique is very similar to that for the DT. The sequential implementation is described as follows:

1) Do DT;

2) Process forward: (using a 3x3 window)

```
For i=2 to M do
    a) For j=2 to N-1 do
        If I(i,j) > 0, then do
        I(i,j)=Max[I(i-1,j-1), I(i-1,j), I(i-1,j+1), I(i,j-1), I(i,j)],     (32)
```

At the end of a), do b)

```
    b) For j=N-1 to 1 do
        If I(i,j) > 0, then do
            I(i,j)=Max[I(i,j+1), I(i,j)];                                   (33)
```

where Max[a,b,..,i] is an operation to find the maximum number among a,b,..,i.

3) Process backward:

```
For i=M-1 to 1 do
    a) For j=N-1 to 2 do
        If I(i,j) > 0, then do
            I(i,j)=Max[I(i+1,j+1), I(i+1,j), I(i+1,j-1), I(i,j+1), I(i,j)],  (34)

At the end of a), do b)
    b) For j=2 to N do
        If I(i,j) > 0, then do
            I(i,j)=Max[I(i,j-1), i(i,j)].                                   (35)
```

4.1.6 Labelling

Labelling is a process to assign an identification number to all pixels in an image. Usually, a same identification number is assigned to all pixels in a common region. Once an image is labelled, the object size, location, and shape can be easily extracted. A labelling technique based on the DT technique was developed and is described as follows:

1) Assign object pixels to "1", and background pixels to "0".

2) Process forward: (using a 3x3 window)

Set label count L=2; and
For i=2 to M do
a) For j=2 to N-1 do
If I(i,j) > 0, then do
I(i,j)=Max[I(i-1,j-1), I(i-1,j), I(i-1,j+1), I(i,j-1), I(i,j)], (36)
If I(i,j)=1, then do
I(i,j)=L, and L=L+1

At the end of a), do b)

b) For j=N-1 to 1 do
If I(i,j) > 0, then do
I(i,j)=Max[I(i,j+1), I(i,j)]; (37)

3) Process backward:

Do the same step 3) specified in 4.1.5;

4) Process forward: (second time)

Do the same step 2) specified in 4.1.5.

4.1.7 Objective Review of Local Operator Performance

The local operators, line detection, image sharpening, and dark and bright object detection were developed and tested using a limited set of images available in Control Data's Image Processing Laboratory and images supplied by RADC for FTCA System Testing. Since there are no suitable criteria for the evaluation of local operators and/or line detection, empirical observations are utilized for checking their performance. In general, they demonstrate performance comparable to the majority of operators found in the available literature. It should be noted that all of the operators developed for FTCA are local operators so the implementation is very efficient.

The local operators described above are used extensively within the OB object detection process described in section 4.7 below. The OB Object detection and counting process when applied to counting cars in parking lots produces a count at the end of processing. Thus the accuracy can be used to measure its performance. Results are mixed. In some cases, the car count is 100 per cent accurate. However, there are several cases that it produces high false alarm rates. In developing the process a target accuracy rate of $\pm$ 30 per cent was established. Based on this goal and the simplicity of the car detector, the performance can be considered acceptable and thus attest to the efficacy of the local operators as well.

Algorithms developed based upon a limited set of test images may not perform as well in practice. As pointed out in [12], automatic target recognition (ATR) algorithms have been partially successful and have produced high false alarm rates in practice. As explained in [12], some of the key reasons for this are the nonrepeatability of the target signature, competing clutter objects having the same shape as the actual targets, experience with a very limited data base, obscured targets, very little use of available information related to and present in the image, such as context, structure, range, etc. Unfortunately, all of those reasons are also true in the case of the FTCA development. On the other hand, those reasons also point out areas for future development efforts to improve performance.

4.2 Automated Comparative Analysis Techniques

In the remaining sections of 4.2 we will provide discussion of the automated comparative analysis techniques within the FTCA System. Included are techniques which utilize combinations of the target area site model, a previously collected reference image, and a new mission image. Various combinations of these 3 elements are used in the different techniques. Some use the site model and new mission image to verify that expected objects are still present. Others use the reference and mission image and apply change detection operators to locate areas of significant change. A general philosophy underlying the approach that we followed in the automated comparative analysis area is that no single technique will work perfectly in all situations. Rather a variety of techniques each performing it's own independent analysis are required. The results of the techniques can then be compared and contrasted with each other in post detection phases. The concept is depicted in figure 4-8.

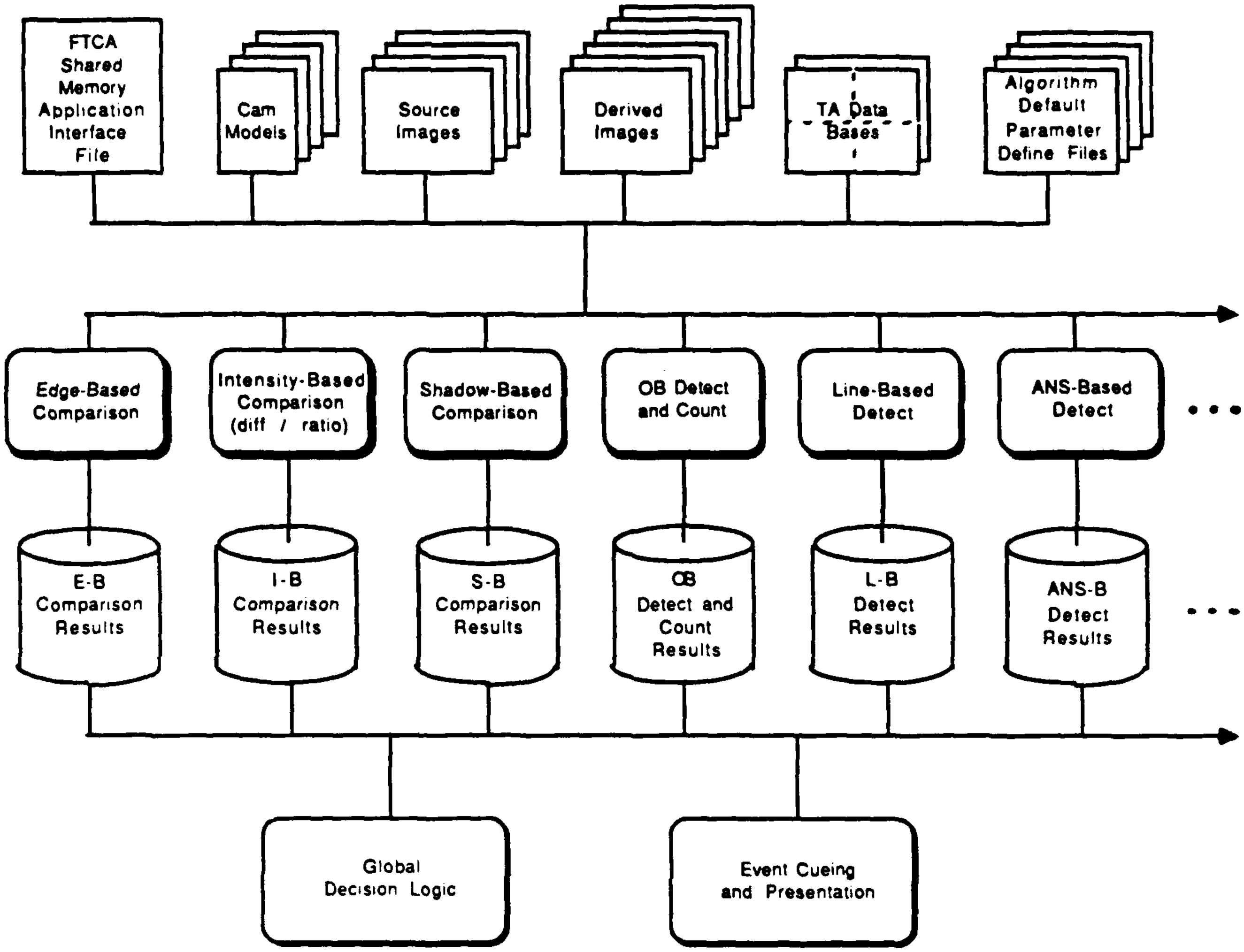


Figure 4-8. Compare function overview.

4.2.1 Site Model Driven Edge-Based Comparison

The EB Comparison technique will only be used on structures which are expected to exist as defined in the Target Area Site Model Data Base. Thus the results can be generated and maintained on a structure-by-structure basis. The processing of 1 structure results in the generation of 1 results record.

The processing attempts to confirm the continued presence of a structure by searching for it's edges. The edges which are searched for are those which are on surface boundaries and are visible as determined by the mission image camera model. Figure 4-9 depicts the concept and figure 4-10 provides an overview of logic. Each "structure results"record contains the results from processing of the structure. Information like expected edge length for all visible segments, maximum and

minimum edge strength found over the search area, search dimension, location of best match, etc. is recorded for each structure. Each structure is either recorded as verified or not verified based upon the how well the expected edge strength matches that found for the set of line segments which are visible for the structure. In addition, preceding all structure records (i,e,. record 0) will be a parameters record which defines some global parameters/constraints commensurate with the processing results.

The event cueing and presentation function (review log option) uses the edge results records to prepare a presentation for cueing the analyst. Each site model object which could not be confirmed is outlined on the mission image. The outline is determined according to the boundary of the site model object as projected for the mission image.

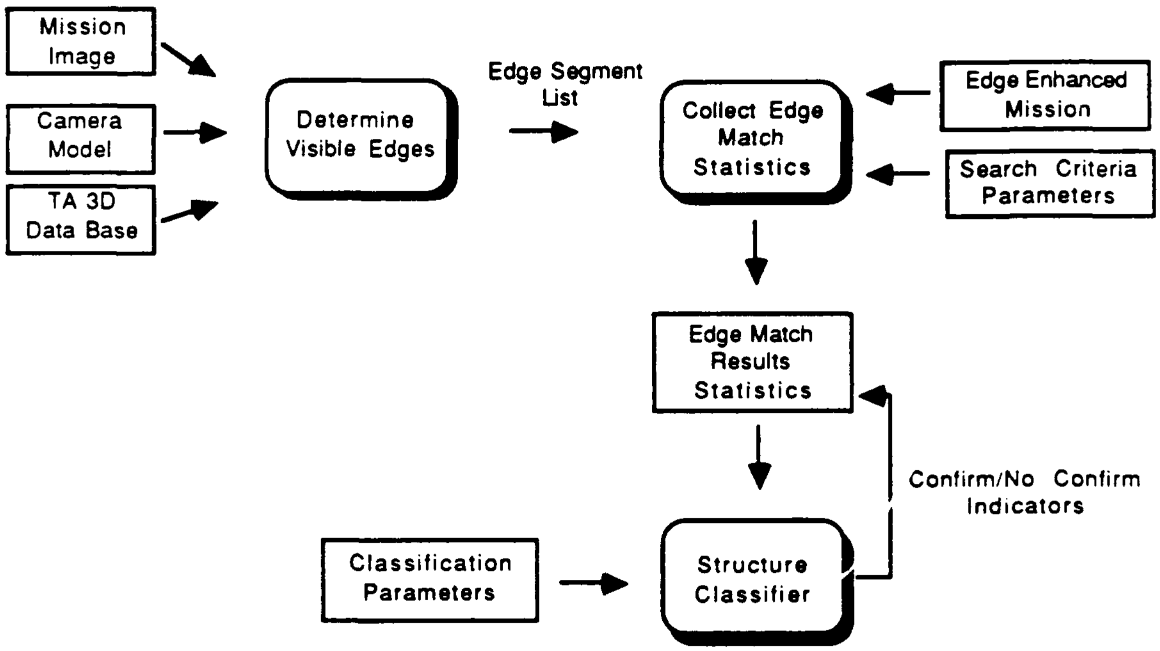


Figure 4-9. Edge-Based verification.

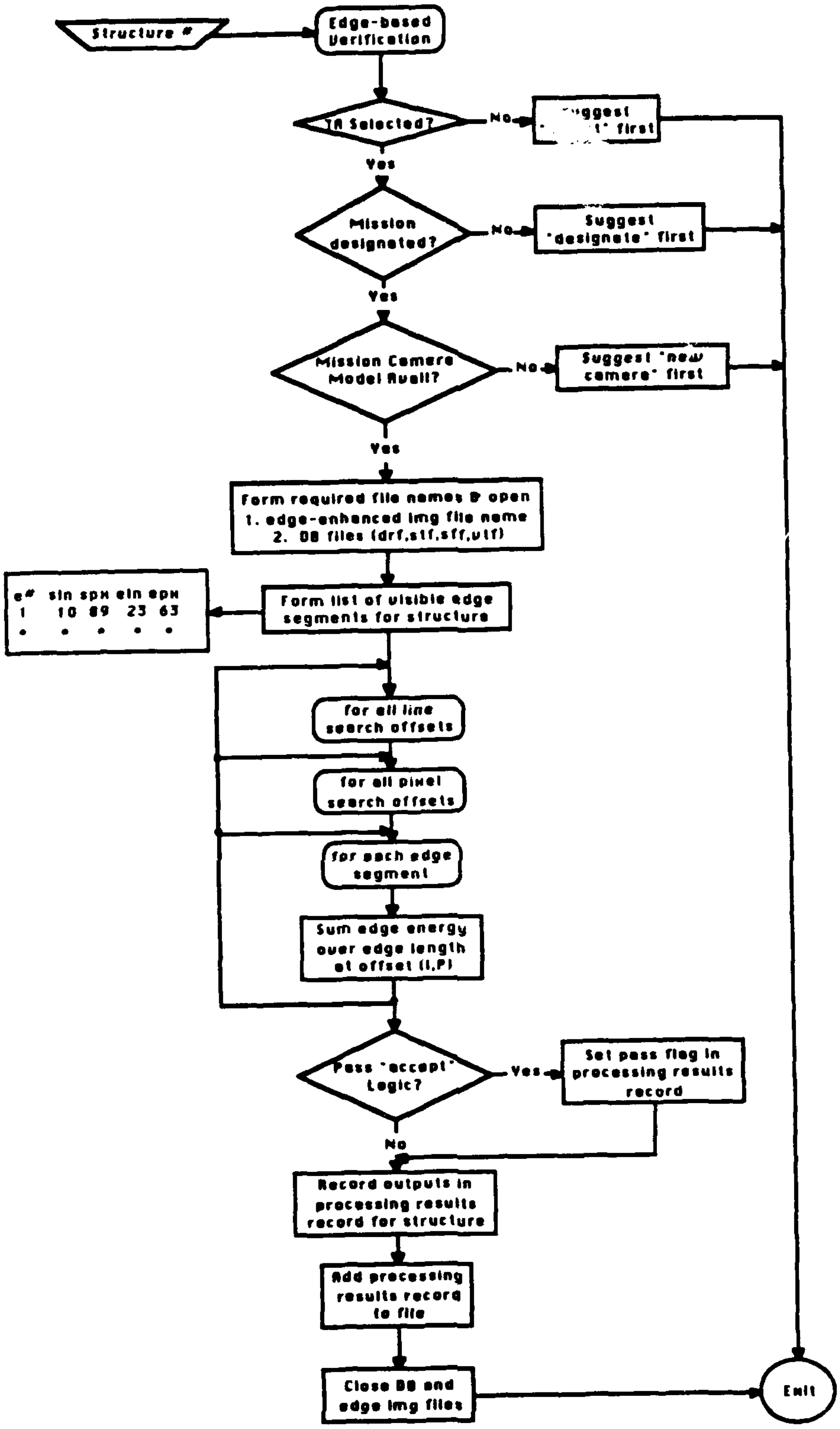


Figure 4-10. Edge-Based verify logic.

4.2.2 Intensity-Based Comparative Analysis

The intensity-based comparison process will attempt to find areas of significant change between the mission and reference images using change detection techniques which compare image intensity data. Figure 4-11 depicts the concept at a high level.

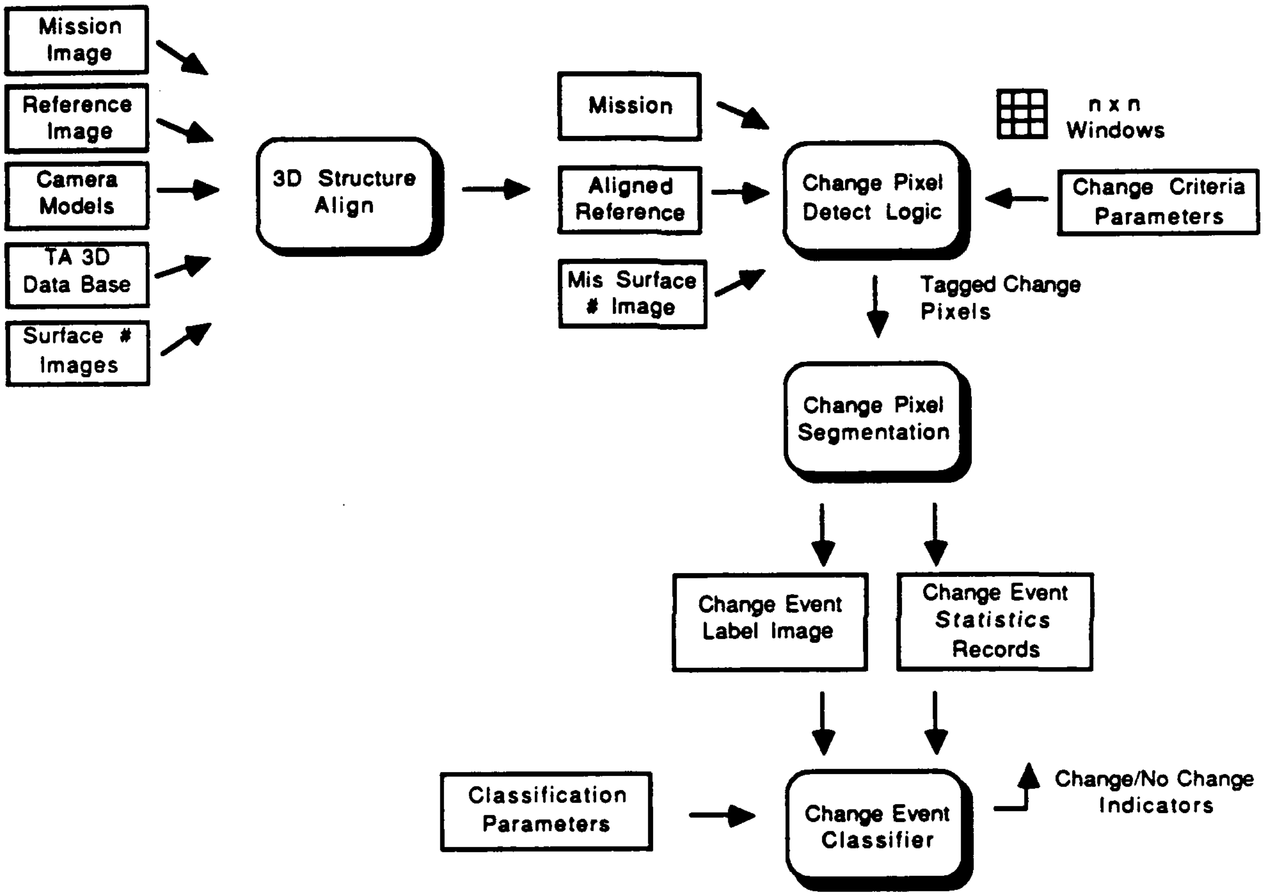


Figure 4-11. Intensity-Based comparison

The first requirement is to have spatially and radiometrically aligned image data. Included within the FTCA System is an image alignment algorithm which utilizes the 3D information in the target area site model to precisely map reference image intensity data so that it spatially aligns with the mission image (described in section 4.5). Surface number images are used to identify planar surfaces in 3 space whose equation can be used in conjunction with the image camera models to precisely define matching locations (line and pixel) in the two images. They also are used to identify places where matching intensity data is not available because of occlusion.

In addition, the images must be brought into global alignment in a radiometric sense (e.g., a large grass field should have approximately the same intensity on both images after correction). A photonormalization process is applied which adjusts the mean and variance of the reference image so that it matches the mission on a global basis. After the aligned reference image is available, a detection process is applied to find areas of change.

The detection logic uses an n x n window (e.g., 3 x 3) surrounding the pixel of interest and does pixel differencing and ratio tests on the intensity data within the n x n window. Initial detection thresholds are applied for determining whether a pixel passes this initial change criteria.

After passing the initial threshold operation all tagged pixels which are adjacent to each other are grouped together to form a candidate change event. During this operation two files are constructed. The first provides a map of the location of change events by writing the event number into an image format for each pixel in the event. In addition, a random access file containing the statistics of the event is produced. This file contains 1 record per event with all of the statistics collected over the event recorded for subsequent use. The statistical parameters collected over each event include size, maximum intensity difference, minimum intensity difference, event boundary rectangle, etc. A complete list of the collected parameters is found in the region.h file in the ftca directory on the system.

Typically at this point a large number of change events have been constructed. In order to eliminate many of the noise-induced events and present only the significant changes to the analyst a classifier is applied to the candidate change events. The statistics in the change event records are utilized by the classifier to determine if each candidate change event is significant or not. If the classifier determines the event is significant, a flag in the change event record is set and each of events classified as change will be presented to the analyst by the event cueing and presentation function.

4.2.3 Line-Based Comparative Analysis

In an aerial image containing man-made objects, linear features such as buildings, and roads are commonly seen. Techniques for the detection and extraction of straight lines and subsequent techniques for comparing those linear features between images provide an additional comparative analysis process.

The line-based detection process is intended as a generic detector. That is, if nothing is known about the area or what might be present, this process can be used to obtain an initial estimate of whether or not the mission image is structurally changed with

respect to the reference image independent of the nature of change. The process detects line segments and compares the 2 line-detected images on a window-by-window basis.

Change windows are recorded for subsequent processing or presentation. Figure 4-12 depicts the concept.

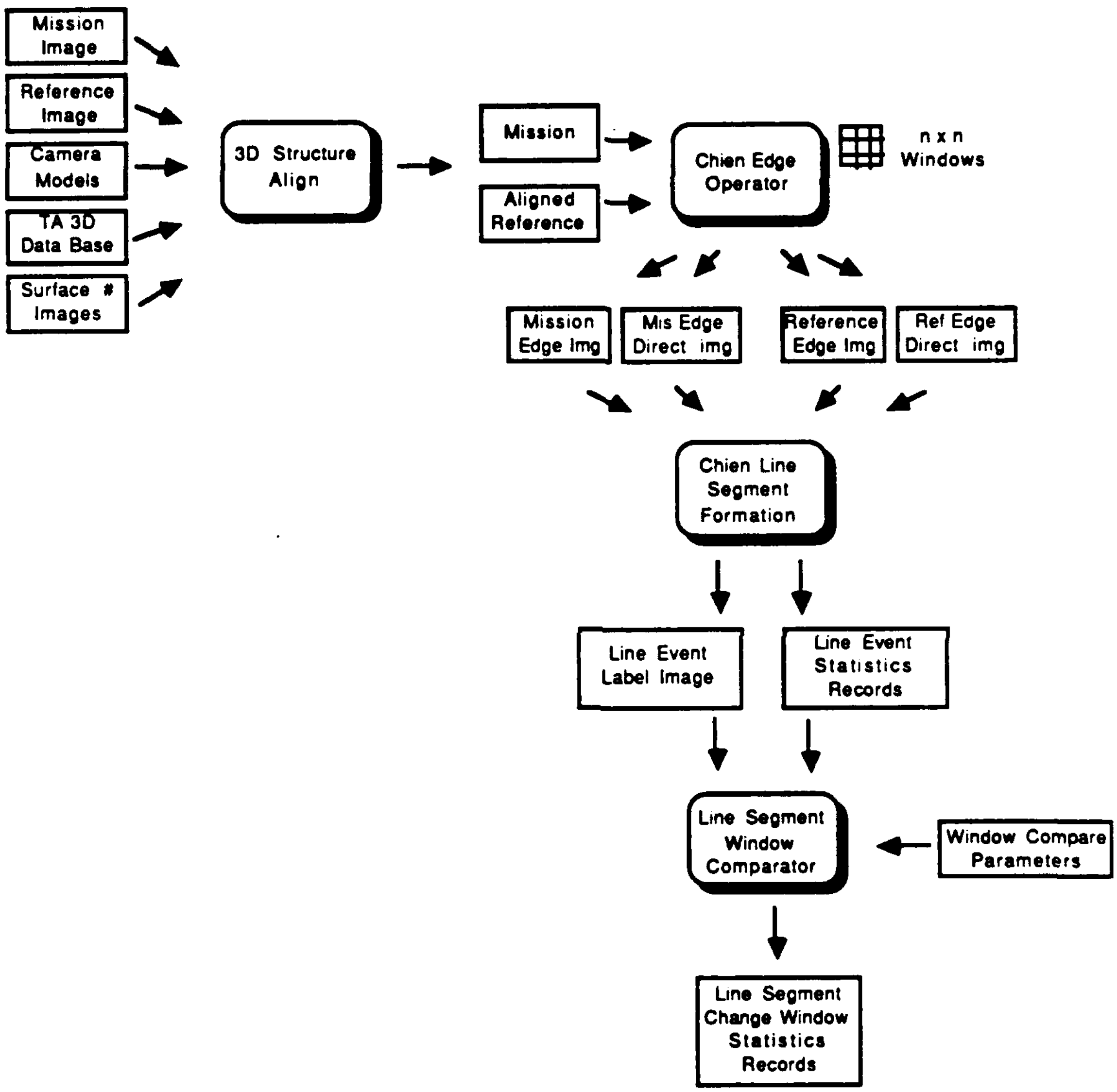


Figure 4-12. Line-Based detection.

There are many techniques for the line detection. Basically, they all require one or more edge operations to assist the extraction of object boundaries. Then, boundary segments are linked and are approximated by a series of piecewise linear segments.

There are two major steps involved in line extraction:

1) Apply an edge operator to produce local information about participating edge elements; and

2) Use line finding techniques to extract line segments.

Edge detection is an important first step for the successful extraction of line segments. It supplies key information for the development of line extraction algorithms. Therefore, a proper selection of an edge operator is critical. However, due to the lack of suitable criteria for the evaluation of edge operators, the choice of an edge operator generally relies on empirical observations. This fact has been clearly indicated in the literature on line extraction. The edge operator used by FTCA was discussed in 4.1.1. After edge preprocessing, the thinning process described in 4.1.1.2.1 is also applied to reduce the number of participating edge pixels. Since line segments do not always have strong edge magnitude, the thresholding process described in 4.1.1.2.2 is skipped in this case.

The next step is to select a line detection technique. After careful study, the use of a Hough Transform (HT) process for extracting straight lines from images was selected. A non-conventional implementation of the HT was developed and will be discussed in the following section.

4.2.3.1 Line Detection Method

The HT is a powerful technique for detecting straight lines in an image. The same concept has been extended to detect circles, ellipses, and other analytically representable curves. It has also been extended to detect other arbitrary shapes or 3-D objects.

The Hough technique can be treated as a maximum likelihood detector. It utilizes a predefined pattern as a template. Participating edge elements vote for all of templates in the Hough space that they belong to. When the vote count of a particular Hough space template exceeds a preset threshold, a template matching is indicated. For example, if the template is a straight line, edge elements are transformed to the Hough space and matched to line templates in the Hough space. Once the vote count of a line template in the Hough space becomes significant, a line with its parameters is extracted.

We made several changes made to the original HT technique as follows:

1) The transformation between the Hough space and image space is eliminated.

2) Line templates are defined in the image space. Thus they are window-type templates. (In the Hough space, they are "points".) The total number of templates is determined by the total number of edge orientation partitions. Each partition specifies a line template. In this implementation, there are 16 partitions. Each partition is assigned an identification number from 1 to 16. The corresponding orientation range is:

   a) If the orientation angle is between $-\pi$ and 0; the range for the partition number N, N=1,..,16 is

   $$(-\pi/16)\,N < \text{angle} \leq (-\pi/16)\,(N-1);$$

   b) If the angle is between 0 and π, the range for the partition number N is

   $$(\pi/16)\,(16-N) < \text{angle} \leq (\pi/16)\,(17-N).$$

   For the extraction of line segments, edge orientations are replaced by a corresponding partition number. Notice that N=0 is reserved for all non-edge pixels. Since the edge orientation is perpendicular to the line direction, the line template is adjusted accordingly. Figure 4-13 depicts all 16 line templates used in for 11x11 pixel windows.

3) The vote count is accumulated in the image space. The accumulation procedure is described as follows:

   a) Check the partition number of the center pixel in a 11x11 window. Proceed to the next pixel if it is 0; Otherwise, project the corresponding line template on the image.

   b) Check the partition number of every image pixel on the projected line template. If its partition number matches or is next to that of the center pixel, the vote count of the center pixel in a corresponding vote-count image is increased by 1.

   c) Repeat steps a) and b) for every pixel in the partition number image.

Notice that since the line template reflects all possible line segments within a range of line slopes specified by the orientation partition through the area covered by the center pixel, two consecutive pixels are sometimes needed to specify a short piece of line in the template. Therefore when both of them contribute a vote to the center pixel, the vote count can only be increased by 1, not 2. The maximum vote count in

a 11x11 window is 10. If the pixel is at the end of a line segment, the maximum vote count is 5. Therefore 5 is considered a reasonable threshold for filtering possible line segments.

The procedure for line extraction can be summarized as follows:

1) Apply the edge operator to a gray-level image, and obtain the edge magnitude and the edge orientation for every pixel in the image.

2) Apply the thinning process to the edge magnitude image, and produce a thinned binary image which specifies participating edge pixels. Figure 4-4b is an example of a thinned binary image. This image will be used as our example.

3) Check the edge orientation of participating edge pixels in the thinned image, replace it by the orientation partition number, and produce a thinned partition number image.

4) Apply the vote count accumulation algorithm discussed above to the partition number image, and produce a vote count image. A vote count image is produced as depicted in Figure 4-14 in which all pixels with a vote count greater than 4 are shown.

5) Apply the maximum value expansion algorithm discussed in 4.1.5 to the vote count image. After applying the algorithm to the image in Figure 4-14, results are depicted in Figure 4-15a in which all pixels with a vote count of 10 are shown. For a comparison, if this algorithm is not applied, results are depicted in Figure 4-15b. Clearly, the difference is significant.

6) Keep all pixels with a vote count of 10. Thus, all line segments longer than 11 pixels are detected.

If the detection of longer line segments is desired, the following steps can be taken after the completion of 6):

a) Apply the labelling algorithm discussed in 4.1.6.

b) Calculate the total number of pixels in every line segment.

c) Keep all line segments with a total number of pixel over a predetermined threshold.

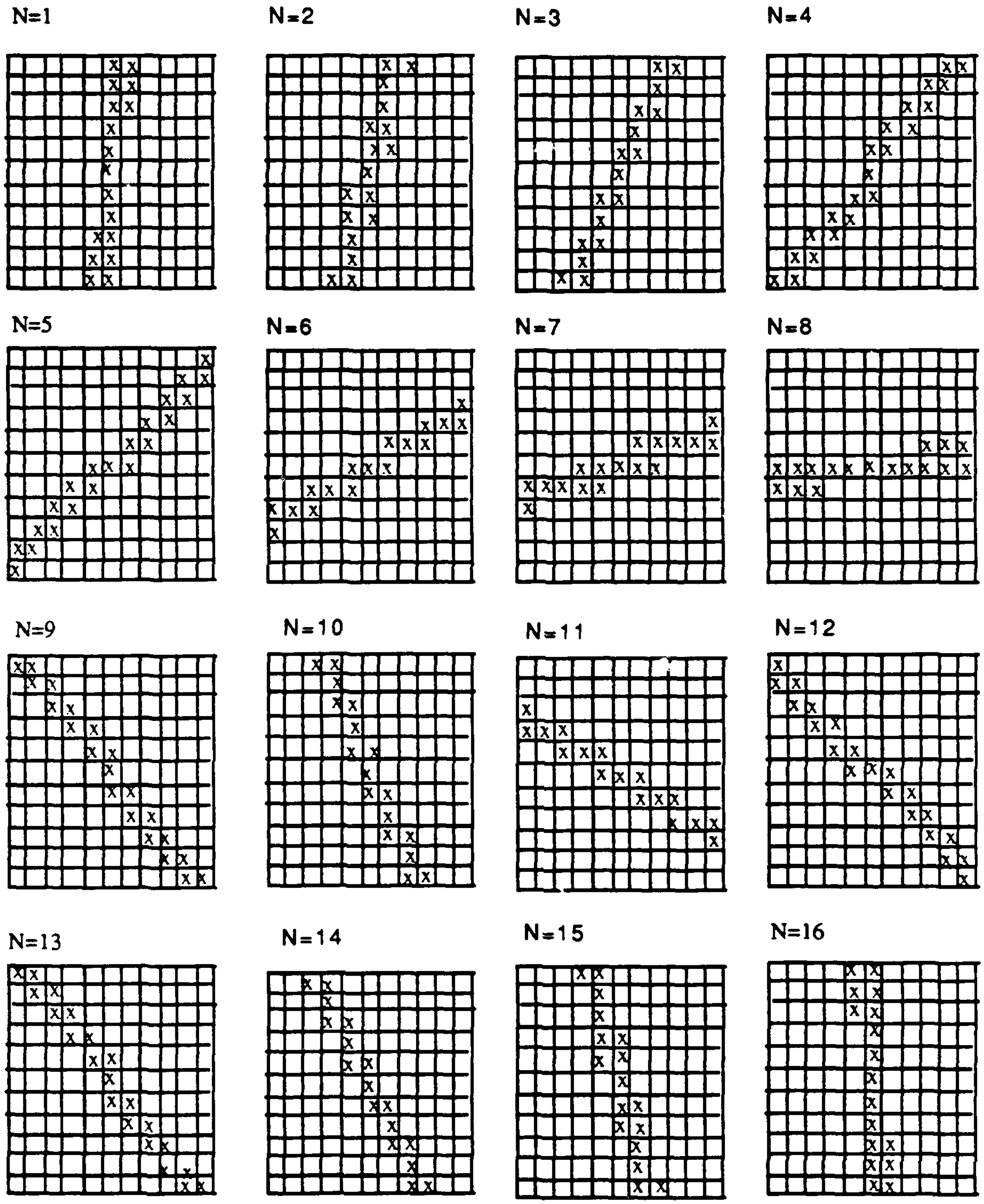


Figure 4-13. Sixteen line templates defined for 11x11 windows.

4.2.3.2 Change Detection Process

After detecting and filtering line segments as described above the resultant line segment images from a mission and aligned reference image are provided to a comparison process. The comparison process is window based. The images are processed as a series of M x M windows (for FTCA testing M=101). The amount of overlap between consecutive windows is selectable. Within each window on the mission and aligned reference the number of locations where a matching line pixel is found in each image is recorded. Each window is shifted with respect to the other within a specified search interval (± 2 for example) to allow for some misalignment between the images. If the count of matching line samples between the 2 image windows is significantly different then the window area is recorded as a change area and is cued for the operator when the comparative analysis results are requested.

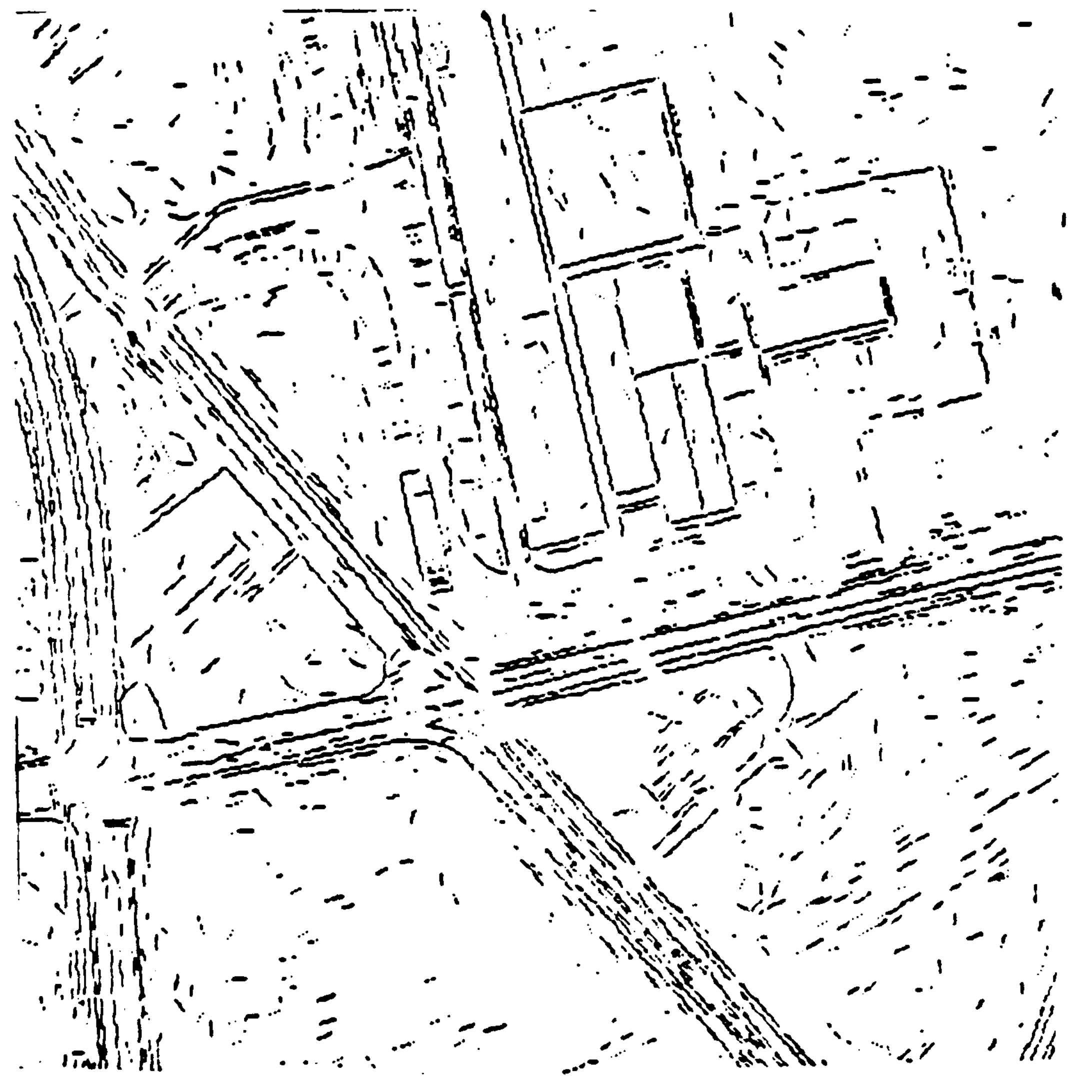


Figure 4-14. Vote count image with vote count greater than 4.

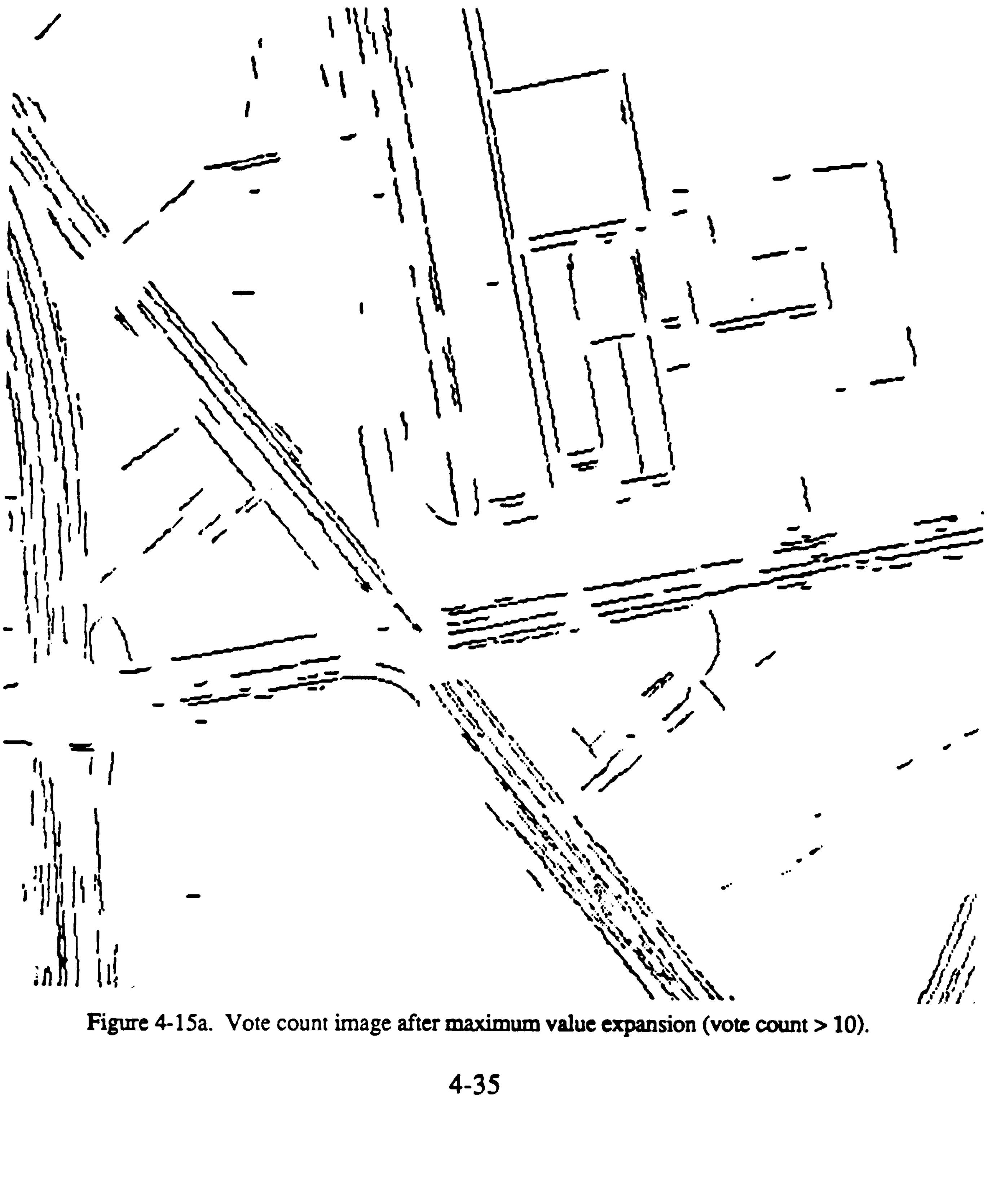


Figure 4-15a. Vote count image after maximum value expansion (vote count > 10).

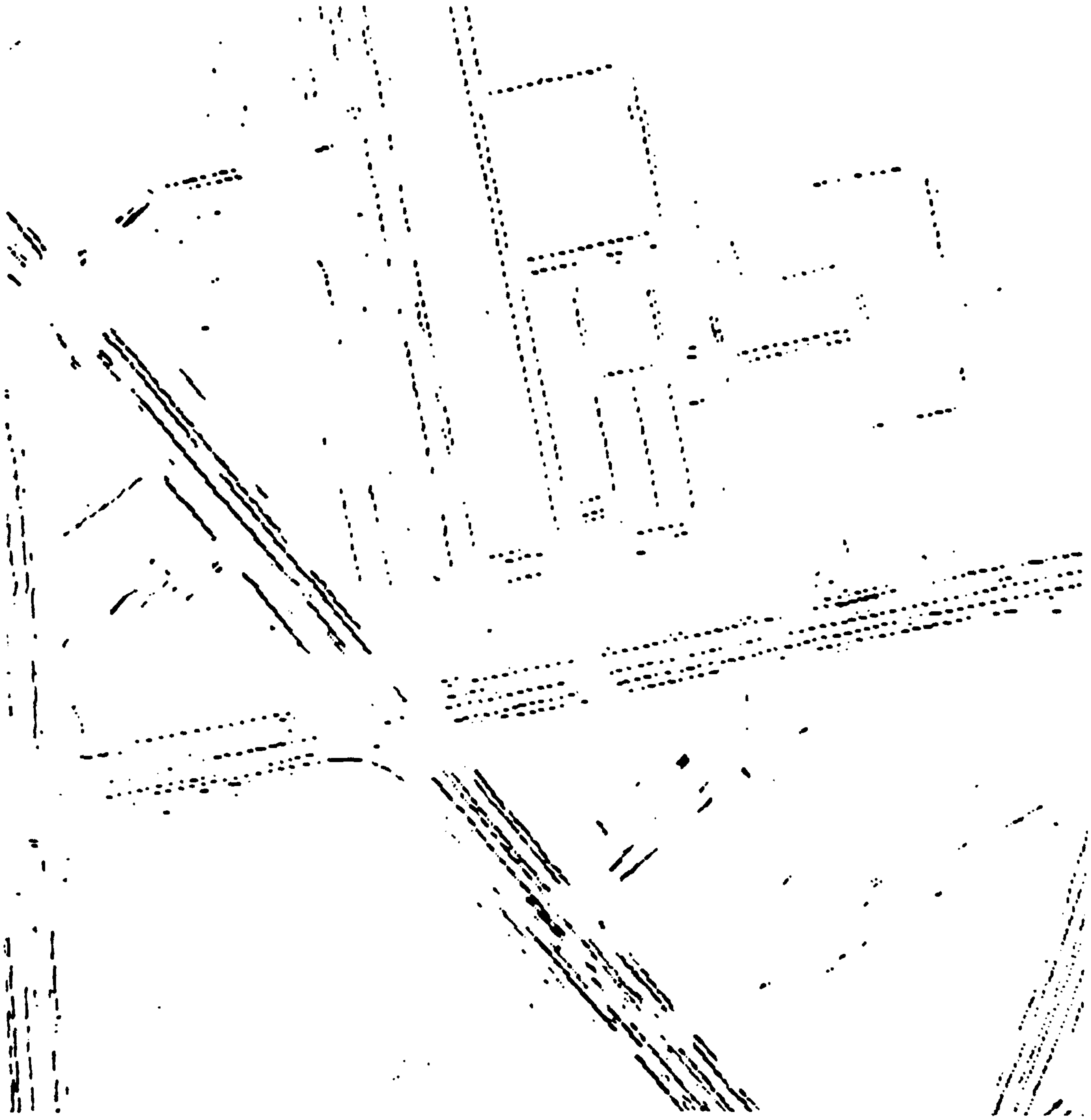


Figure 4-15b. Vote count image no MV expansion (vote count > 10).

4.2.4. Shadow-Based Comparative Analysis

The shadow-based comparison process attempts to locate shadows in the mission image and determine whether or not they can be associated with existing structures in the target area site model data base. If they can, they are assigned the structure's id. If they can not and are of sufficient size and character, they represent possible new changes. Thus there are two aspects to the shadow-based comparative analysis processing; a verification process which confirms expected objects, and a new object detection process. The data flow for the concept is depicted in figure 4-16.

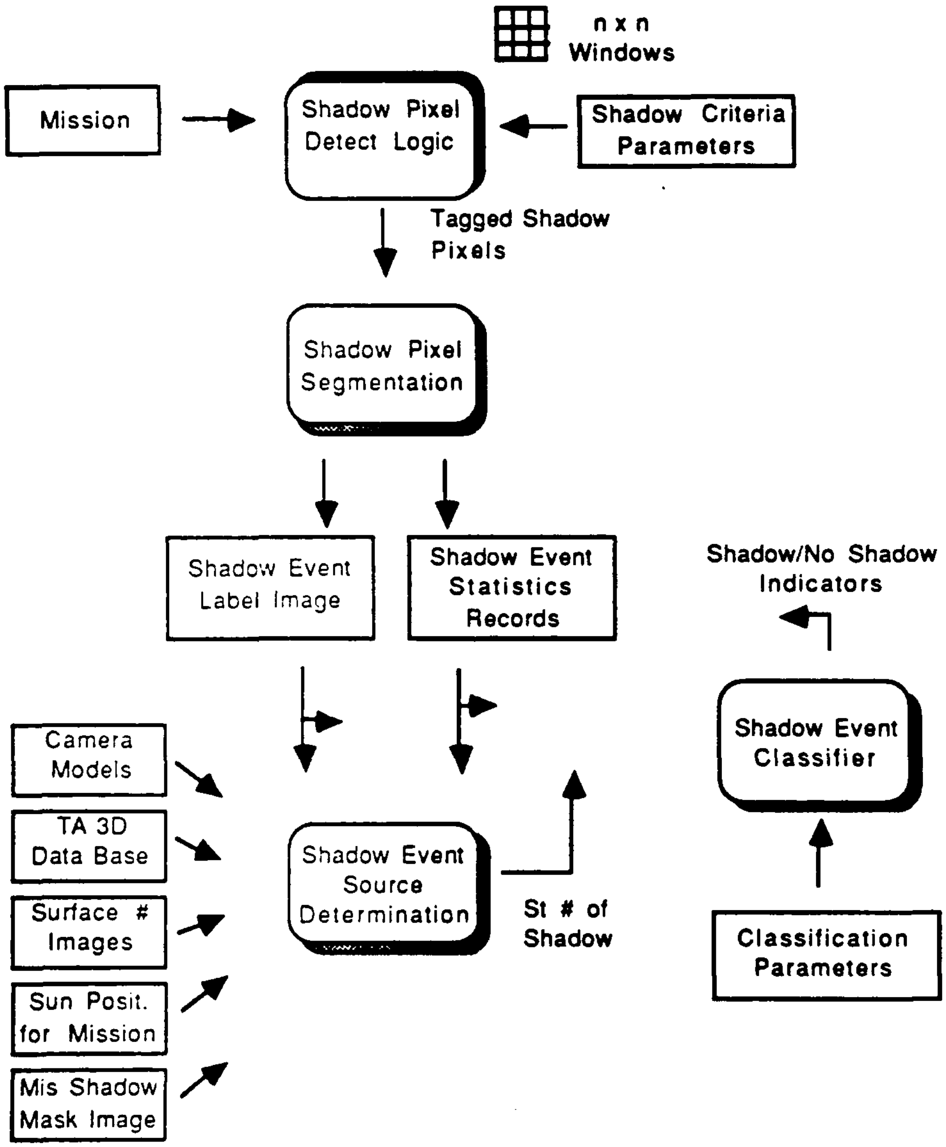


Figure 4-16. Shadow-based comparison

4.2.4.1 Site Model Driven Shadow-Based Verification

The shadow-based verification process uses the site model and mission image as it's basis for comparison. There are 3 components to the shadow-based verification process as follows:

- Generate predicted shadow image
- Detect likely shadow pixel of mission image
- Compare predicted against detected to confirm objects

To generate the predicted shadow image the direction to the sun (azimuth and elevation angles) is required for the mission image. A sun position surface map image is generated to record which surfaces are visible from the sun's perspective. This image is compared to the mission image surface number image and when they are not in agreement this indicates a shadow pixel and the structure causing the shadow is recorded as the intensity in the shadow predicted image.

The shadow detection process described in section 4.1.3 is applied and the shadow detection image is produced. This is simply a binary image which marks all pixels which are likely to be shadows.

The shadow-predicted and shadow-detected images are compared and for each site model object a count of predicted versus detected shadow pixels is accumulated. If the number does not match within a selected tolerance, the object is recorded as not confirmed and the results presentation and cueing function will provide the analyst with a cue for each non-confirmed site model object.

4.2.4.2 New Event Detection from Shadows

In addition to confirming the existence of expected objects in the site model, shadow processing can also be used to detect new objects of interest. The shadow-detected mission image is submitted to a segmentation and labelling process to form shadow regions. A number of description parameters is recorded for each region to produce a feature vector which can be used by a classification program. The complete list of collected statistical parameters is found in the region.h file in the ftca directory. The feature vector of each shadow region is submitted to the shadow event classifier which determines if the region is of interest or not. If determined to be interesting, the logic checks to see if the region is associated with an expected site model object. If it is, no change record is generated. If it is not, a change record is generated. The results presentation and cueing function will provide the analyst with a cue for each shadow region recorded as a shadow-induced change of interest.

4.2.5 Artificial Neural System Comparative Analysis

The FTCA System includes an Artificial Neural System (ANS) implementation for finding areas of change between 2 aligned images. The required processing components to accomplish this are as follows:

- training data selection and preparation
- training
- application for change detection.

In general we will attempt to use a neural net implementation to determine whether or not 2 square windows (~ 64 x 64) of image data (One from aligned reference and one from mission) are the same or not. Prior to discussing each of the above processing components details on the similarity measure we used for classification and the resultant network architecture will be provided. The sections following that will provide the implementation details for each of the above processing components.

4.2.5.1 Similarity Measure

Given two patches we now would like to define a vector A, whose components are a computable function of the patches. Furthermore, we would like A to be sufficient for classification of the patches. That is, we must also define a function F(.) such that F (A) > 0, implies that patches are similar, and F(A) < 0, dissimilar Moreover, we would like to make the same decision if the ordering of the patches is interchanged or if each patch is rotated by the same angle. The components of A are called features, and F(.) is called the decision function.

The metrices used for classification are intensity moments. They, along with a correlation metric are used in the implementation. Intensity moments are defined here as

$$m_{ij} = \sum_{P} x^i y^j I(x,y)$$

where (x,y) are the coordinates of a pixel with respect to a coordinate system whose origin is the patch's center, I(x,y) is the intensity at (x,y), P is the square patch under consideration, and i,j are non-negative integers.

Here we let (i,j) = 0, 1, 2, 3. This yields 33 metrices (16 moments from each patch and the correlation metric) which somehow must classify the patch pair. However, these raw metrices cannot be used directly as components of A as explained below.

After registration, the two transformed images are "photonormalized". That is, one of the registered images has its intensity transformed as J = aI + b. Here I is the original intensity at any pixel, J is the new intensity at that pixel, a and b are constants chosen so that the mean and variance of the two final images are the same. Thus the m_{00}'s are not useful for classifications.

We might take the m's from each of the two patches to form the components of A. But we wish the classification to be the same if the numbering of the patches is interchanged and if each patch is rotated the same angle. With these restrictions in mind, the components of A are formed as:

$$A = (r_{01} + r_{10}, r_{02} + r_{20}, r_{03} + r_{30}, r_{11}, r_{12} + r_{21}, r_{13} + r_{31}, r_{22}, r_{23} + r_{32}, r_{33}, \text{correlation}, 1) \qquad (2)$$

hence,

A is a 1 x 11 vector,

$$r_{ij} = \frac{m_{ij}}{M_{ij}} + \frac{M_{ij}}{m_{ij}},$$

m_{ij} is the (i,j)th moment from the first image patch, and

M_{ij} is the (i,j)th moment from the second image patch.

Note that if the patch numbering is interchanged or if each patch is rotated $\pm 90^\circ$ or 180°, then A is unchanged. Next the decision functions, F(.), must be considered. This is supplied by the so called "back propagation neural net".

4.2.5.2 Neural Net Classification Architecture

At this point we have a 1 x 11 vector A, formed from the intensity moments of the two patches, their correlation, and a constant = 1. We hope that some function of A will correctly classify the patches. The functional form chosen is that of the neural net paradigm called backpropagation [14, 15]. Even though this paradigm is discussed in the literature, we will provide an overview. This can be done since it is quite simple, if biology and philosophy issues are omitted.

The decision function to be used is as follows:

$$B = f(AU)$$

$$F(A) = AV + BW = AV + f(AU)W, \tag{3}$$

where:

A is a 1x11 vector (input vector)
U is a 11x3 matrix
B is 1x3
V is 11x1
W is 3x1
$f(x) = 1/(1+e^{-x})$
f(matrix) = matrix composed by applying f(.) to each element of the matrix.

Given the 47 constants which determine the matrices U, V, and W; F(A) can be computed for any A. Now U, V, W must be chosen so that $F(A) > 0$ implies the patches which determine A are similar, and $F(A) < 0$ implies dissimilar.

In the language of the backpropagation paradigm, U, V, W are called weights. Equation (3) is said to represent an input layer with 11 nodes, a hidden layer with three nodes and one output node. A diagram as shown in Figure 4-16 can also be used to depict the network. Here only 3 of the 47 connections are shown.

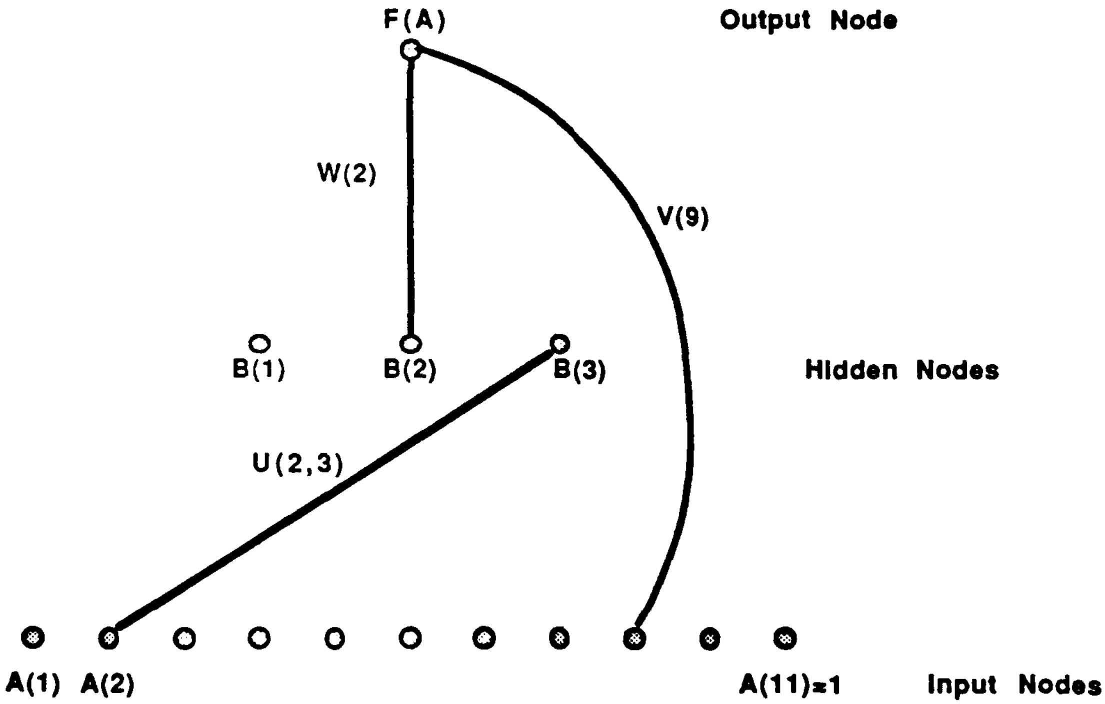


Figure 4-16. Neural Net Corresponding to Equation (3).

The method of obtaining U, V, and W is called "training". This is done by use of several A's, each of which is of a known class. This will be explained in the next section, but first we should explain the definition of B in (3) and why the last component of A is 1.

Augumenting the vector provided by the computed moments and correlation simply provides more flexibility to the decision surface than otherwise obtained. That is, the decision surface F(A) = 0, is a surface in a 10 dimension feature space. A less restricted surface is obtained by allowing 4 additional weights in its definition.

Some type of nonlinearity is required in the definition of B, otherwise, the decision surface would simply be a plane in the 10 dimension feature space; a more general surface is required. Why the particular f(.) is chosen by the paradigm (the so called "sigmoid function") is not clear. Indeed chosing $f(x) = x^2$ may well be more appropriate to some situations.

4.2.5.3 Training Data Selection

To train the network it is necessary to select examples of patches on the aligned images which represent change and no-change cases. FTCA provides such a capability in the form of an interactive visual interface which allows the analyst to point at windows on the images and tell the system whether they are change or no-change cases.

After all training windows have been identified, the system computes and strores the feature vector for each window pair as defined above. The training data file can be extended if more training samples need to be included to improve performance after classification results are available.

4.2.5.4 ANS Training

It is now required to determine the 47 unknowns which define U, V, W. This is done by using the training data and seeking the U, V, W which define F(.) such that the decision function is correct or nearly correct over this set of patch pairs. It is then hoped that this so obtained function will correctly classify any general patch pair.

Let p patch pairs of known classification be provided in the training data file. Then form a px11 matrix, C, and px1 vector, T, as follows:

$$C = \begin{pmatrix} A(1) \\ A(2) \\ \bullet \\ \bullet \\ \bullet \\ A(p) \end{pmatrix}, \quad T = \begin{pmatrix} T(1) \\ T(2) \\ \bullet \\ \bullet \\ \bullet \\ T(p) \end{pmatrix}$$

A(i) is formed from the ith pair of the training set
T(i) = -1 if the ith pair is dissimilar
= 1 if the ith pair is similar
also let,

$$D = CV + f\,(CU)W \tag{4}$$
$$E\,(U,V,W) = (D-T)^T\,(D-T). \qquad (X^T \text{ is the transpose of } X)$$

Now, chose U, V, W so that E is as small as possible. Note that E is a differentiable scalar function of 47 variables. The minimization of such a function is a standard problem in numerical analysis. A conjugate gradient method [16] may be used.

4.2.5.4.1 Simple Example of ANS Training

To make this more clear we will consider training for a simple two class problem. Consider a simple problem with p=14, only 2 features instead of 10, and only one hidden node instead of 3. Since there are only 2 features, the decision curve can easily be sketched. Let the T's and the first two components of the A's be given in the following table. Here, the feature vector has the form $A = (A_1,A_2,1)$.

A_1	A_2	T
1	7	-1
6	3	-1
4	7	-1
5.5	6.5	-1
3.5	8.5	-1
5	9	-1
3	11	-1
7.5	4.5	1
7	6	1
6.5	7.5	1
6	9	1
4	12	1
5.5	11.5	1
9	13	1

From eq. (4) $D = CV + f(CU)W$, and

$$E = (D-T)^T (D-T)$$

where U, V, W are respectively 3x1, 3x1, and 1x1. So E is a function of seven variables. Minimizing E over these variables yields

$$U = \begin{pmatrix} .845302 \\ .75598 \\ .00510 \end{pmatrix}, \quad V = \begin{pmatrix} .12159 \\ .34149 \\ -4.13963 \end{pmatrix}, \quad W = 2.34416.$$

The decision curve is then

$$0 = V_1 A_1 + V_2 A_2 + V_3 + W f (U_1 A_1 + U_2 A_2 + U_3).$$

Figure 4-17 shows the graph of this curve (called linear sigmoidal) and the 14 training points. Note that the curve does indeed separate the two classes. However, points 7 and 11 are barely on the correct sides.

After using this formulation on this simple problem, an important defect may be noted. The unknowns were determined as a least-squares solution. This means that points 1 and 14 were given more weight in determining the decision curve than any of the other points. However, this makes little sense since they are clearly easy to classify. The more critical points should have a greater weight and not a lesser weight as this formulation prescribes. This inverse weighting defect is removed by the neural net paradigm.

Let the two following changes be made. Modify (4) so it becomes

$$D = f(CV + f(CU)\ W) \tag{5}$$

and change the Class 1 components of T to 0 (formerly they were -1). With these changes, points far removed from the decision surface will have a small weight in determining the minimum value of E. That is, suppose a vector A_i is far from the decision surface, $F(A) = 0$, thus $F(A_i) >> 0$ or $F(A_i) << 0$. Now, if the sigmoidal function is used in the definition of D, the contribution to E (eq. (4)) due to A_i is very small. But if D were defined without the sigmoidal function, then its contribution to E is very large. Thus using the sigmoidal function in the definition of D forces the decision surface to be influenced more greatly by feature vectors close to the surface than those far from the surface. Use of the sigmoidal function in the definition of D (Equation (5)) is clever. The decision surface is still given by $F(A) = 0$ where F(.) is defined by (3).

A price must be paid for this change in the definition of D however. Now V and W enter the minimization problem nonlinearly. When (5) is used instead of (4) for the definition of D, then minimizing E yields

$$U = \begin{pmatrix} -.19023 \\ -.08349 \\ 1.72615 \end{pmatrix}, \quad V = \begin{pmatrix} 2.42975 \\ 1.03853 \\ -19.96314 \end{pmatrix}, \quad W = -3.15857$$

Figure 4-17 shows the graph of this curve (called Sigmoidal-Sigmoidal). Note that the more logical weighting provided by the final sigmoidal function separates the classes better than the linear weighting.

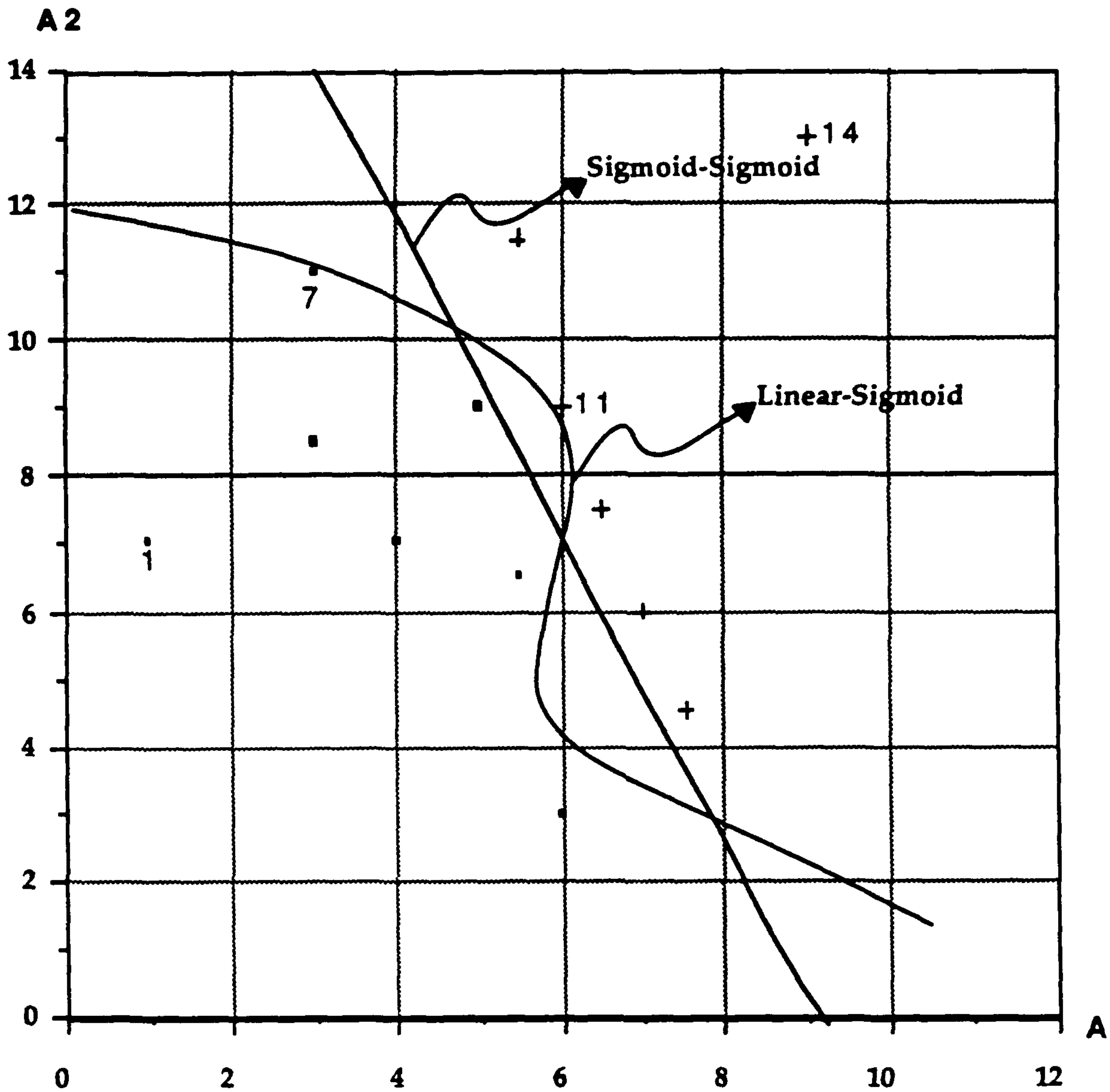


Figure 4-17. Linear-Sigmoidal and Sigmoid-Sigmoid Decision Curves for Simple Problem.

4.2.5.5 Training For Image Classification

Two pairs of aligned images were used for training and network evaluation during program development. The aligned images are shown in figure 5-5.

From these images, 320 square patch pairs were selected. Each patch was 64x64 pixels, and each patch pair was manually classified. A feature vector was computed

for each patch pair as defined by Eq. (2). As a result of this training the following weights were obtained:

$$U = \begin{vmatrix} -.635 & .430 & -.155 \\ .618 & -1.186 & -.078 \\ .475 & .389 & -.187 \\ .543 & .619 & .054 \\ .420 & -.087 & -.433 \\ -.043 & .712 & -.565 \\ .603 & -1.018 & .633 \\ -.403 & .879 & .200 \\ -.584 & -.386 & -.805 \\ -2.44 & -2.341 & -2.174 \\ .770 & -1.027 & .018 \end{vmatrix}, V = \begin{vmatrix} .021 \\ .057 \\ .159 \\ .114 \\ .019 \\ -.039 \\ .136 \\ -.309 \\ -.505 \\ 9.080 \\ .679 \end{vmatrix}, W = \begin{vmatrix} -4.016 \\ -6.309 \\ -8.237 \end{vmatrix}$$

After training, 7 of 320 cases were misclassified by the trained net.

At this point, we have a decision surface in 10-space. A rough view of this surface may be obtained if 8 components of A are fixed and the curve obtained by allowing the remaining two components to vary is sketched. First we chose

$$A = (4, 4, 4, 2, 4, 4, 2, 4, A_9, A_{10}, 1)$$

then, sketch the decision surface (curve in this case)

$$0 = AV + f(AU)\,W \text{ as a function of } A_9 \text{ and } A_{10}$$

The result is shown in Figure 4-18. From eq. (2) we obtain A_9 from m_{33} and M_{33}, and A_{10} is the correlation metric.

Note that from our definition of the feature vector A, if the two patches were identical then A = (4, 4, 4, 2, 4, 4, 2, 4, 2, 1, 1). So the point $(A_9, A_{10}) = (2,1)$ must clearly be on the no-change side of the decision curve in Figure 4-18. This is indeed the case.

Next, we choose $A = (0, 0, 0, 0, 0, 0, 0, 0, A_9, A_{10}, 1)$ and sketch the decision curve, also shown in Figure 4-18. Since the first 8 components are now more favorable to a change, the change region is increased. Note that if A_{10} (the correlation metric) is <.3, a change is always indicated. This shows that correlation is a strong indicator of

change or no-change. However, the other components of A are also important indicators.

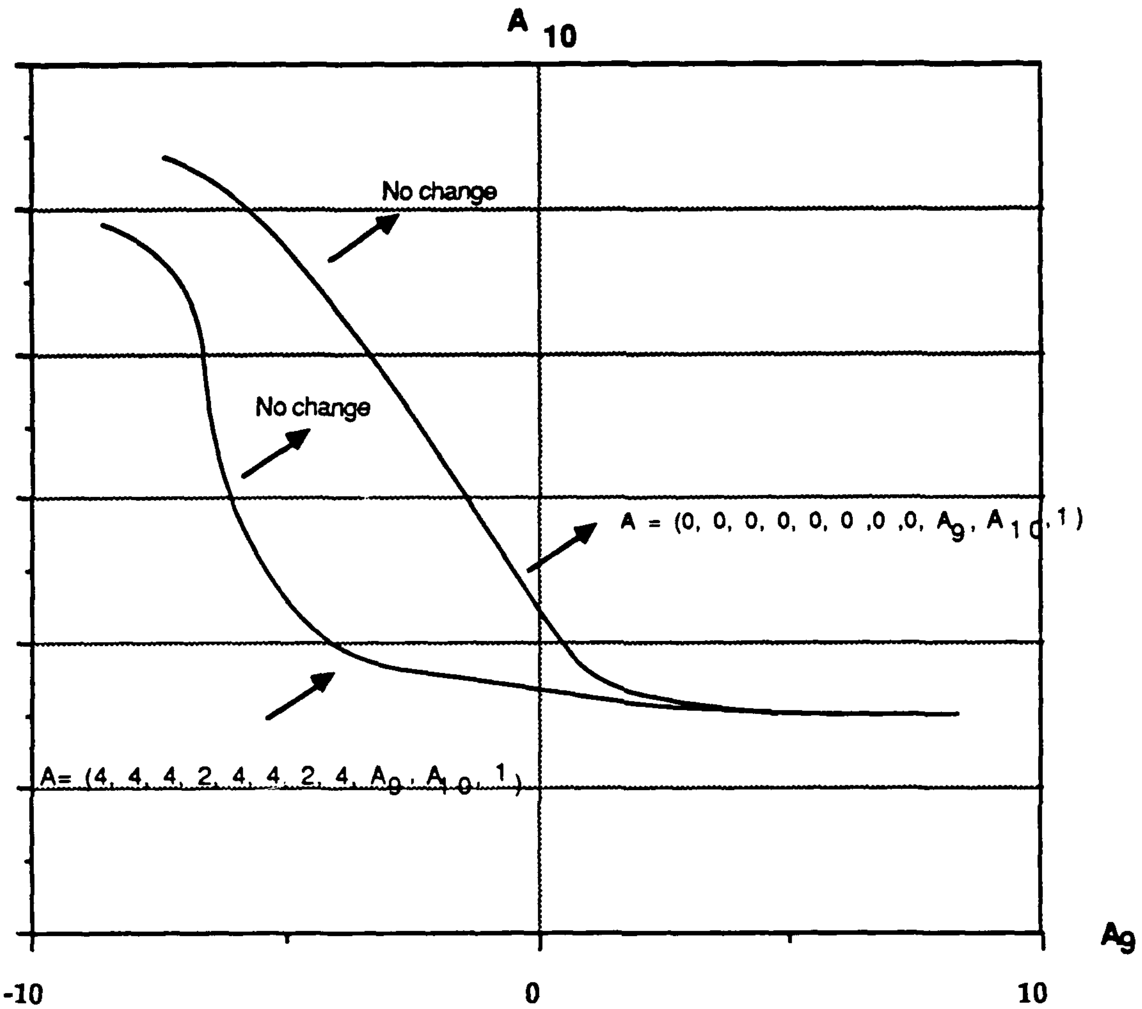


Figure 4-18. Decision Curve as a Function of A_9 and A_{10} for fixed $A_1, \ldots, A_8$.

4.2.5.6 Nominal Test Results

During development the network was tested after it was trained to assess how well it could be expected to perform. Using the registered test images 160 new test samples were selected. They were once again each 64 x 64 pixel patches. Each patch pair was classified by the decision function with the given numerical values of U, V, and W.

From 160 cases, 2 were incorrectly classified as no-change, 9 were incorrectly classified as changes. This yields a probability of detection of .975, and a false alarm rate of .056. This seems to be nominally the performance noted on the imagery used for system testing as well.

4.2.5.7 Application for Comparative Analysis

In general, after training the network for a target area, it can be applied to a pair of aligned images to identify locations of change. In order to do this the entire registered images are partitioned into contiguous 64 x 64 pixel patches. Each patch pair is classified by the decision function with the numerical values of U, V, and W which resulted from training.

Each time a window is determined to be dissimilar a change record is produced. The results presentation and cueing function will provide the analyst with a cue for each window recorded as indicating change.

4.3 OB Object Detection and Counting

The FTCA system included a requirement to detect and count OB objects. The delivered capability allows the detection and counting of a fixed set of man-made objects. The analyst can predefine detect and count areas and object type desired or can interactively specify a region over which detection and counting is to be performed.

The discussion of the technique will use cars as an exemplary object for detection and counting. In an aerial image containing man-made objects, a car can be considered as a small object. Its size is unique enough that by checking an object's size alone, a car can be detected with good accuracy. This property is utilized for the car detection and counting.

The major problem is how to enhance an image so that a small object with an expected size can be detected and subsequently counted. If an object has uniform intensity which is significantly different from the background intensity in an image, the detection of this object is relatively easy. In this case, all detected edge contours are object boundaries. So, the object is well defined. On the other hand, if an object has several small dark and bright regions which are also significantly different from the background intensity in an image, the detection of this object is more complex. Now, all detected edge contours are a mixture of object boundaries and region boundaries within the object. Since object and interior region edges appear similar

in the edge image and furthermore are not always well connected, it is much harder to classify and properly merge them to form a single object.

Notice that a dark car is generally a dark and uniform-intensity object. But, a bright car is an object with dark and bright regions: thin shadow, windshield, and rear window (if it is visible) may form dark regions; hood, top, and trunk (or cargo bed) may form bright regions. Thus both cases mentioned above apply to cars and other OB objects of interest. Image enhancement techniques for improved detection in light of these considerations will be discussed in the following section.

4.7.1 Small OB Object Enhancement

The background around a car usually has a gray-level intensity in between the dark and bright intensity of a car. Furthermore, intensity changes between the background and the object are significant. If the background intensity is used as a reference and any intensity change above or below the reference is treated equally, then the edge strength of bright and dark pixels in a car can be equally emphasized. In this case, the edge strength contour between dark and bright region is eliminated. However, the edge strength contour between the background and the object is still visible in the edge strength image.

A easy way to accomplish this, is described as follows:

1) Search for the maximum (Imax) and minimum (Imin) intensity in a 3x3 window.

2) Use equations (17) and (18) in 4.1.2.1 to calculate DImax and DImin, respectively. As discussed in 4.1.2.1, they measure the darkness or brightness of the center pixel, respectively.

3) If DImax > DImin, assign DImax to the center pixel of the window; Otherwise, assign DImin to the center pixel. If DImax > DImin, this indicates the center pixel is a dark pixel. So, the measure of the darkness is a proper edge strength indicator for the center pixel.

4) Repeat steps 1) to 3) for all pixels in the image, and produce an edge strength image.

5) Threshold the edge strength image using the thresholding technique described in 4.1.1.2.2, and keep all pixels above the threshold. Figure 4-19a), b) and c) depict a thresholded edge strength image with dark pixels, bright pixels, and all pixels, respectively.

The major drawback of this method is that all edge boundaries are extended one pixel into the background. This will merge two closely-spaced objects together. If a car is parked near a building, near the boundary between two homogeneous regions, near a shadow area, or too close to another car in a parking lot, erroneous merges are quite likely to happen. A solution for this problem will be addressed in the following section.

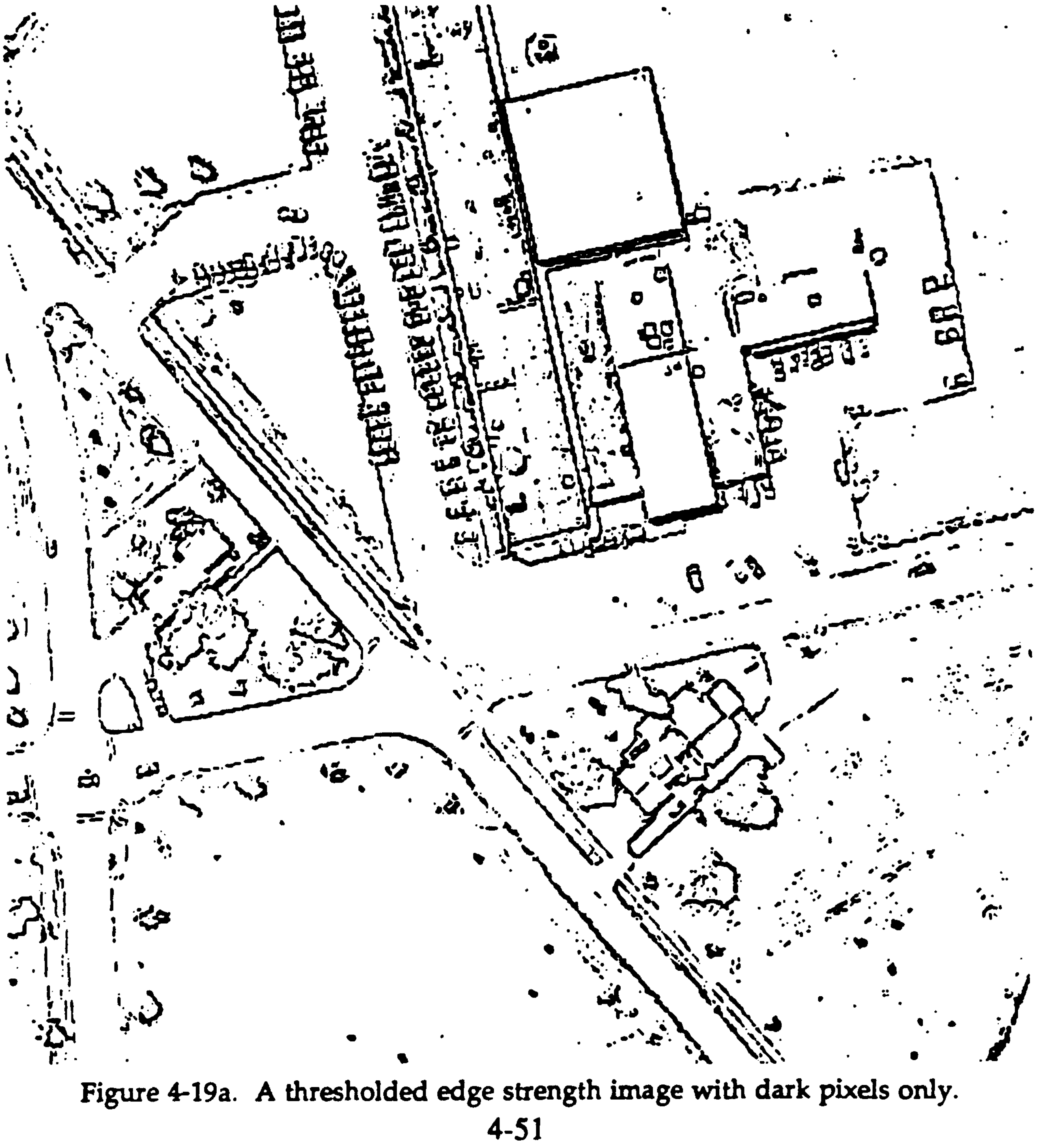


Figure 4-19a. A thresholded edge strength image with dark pixels only.

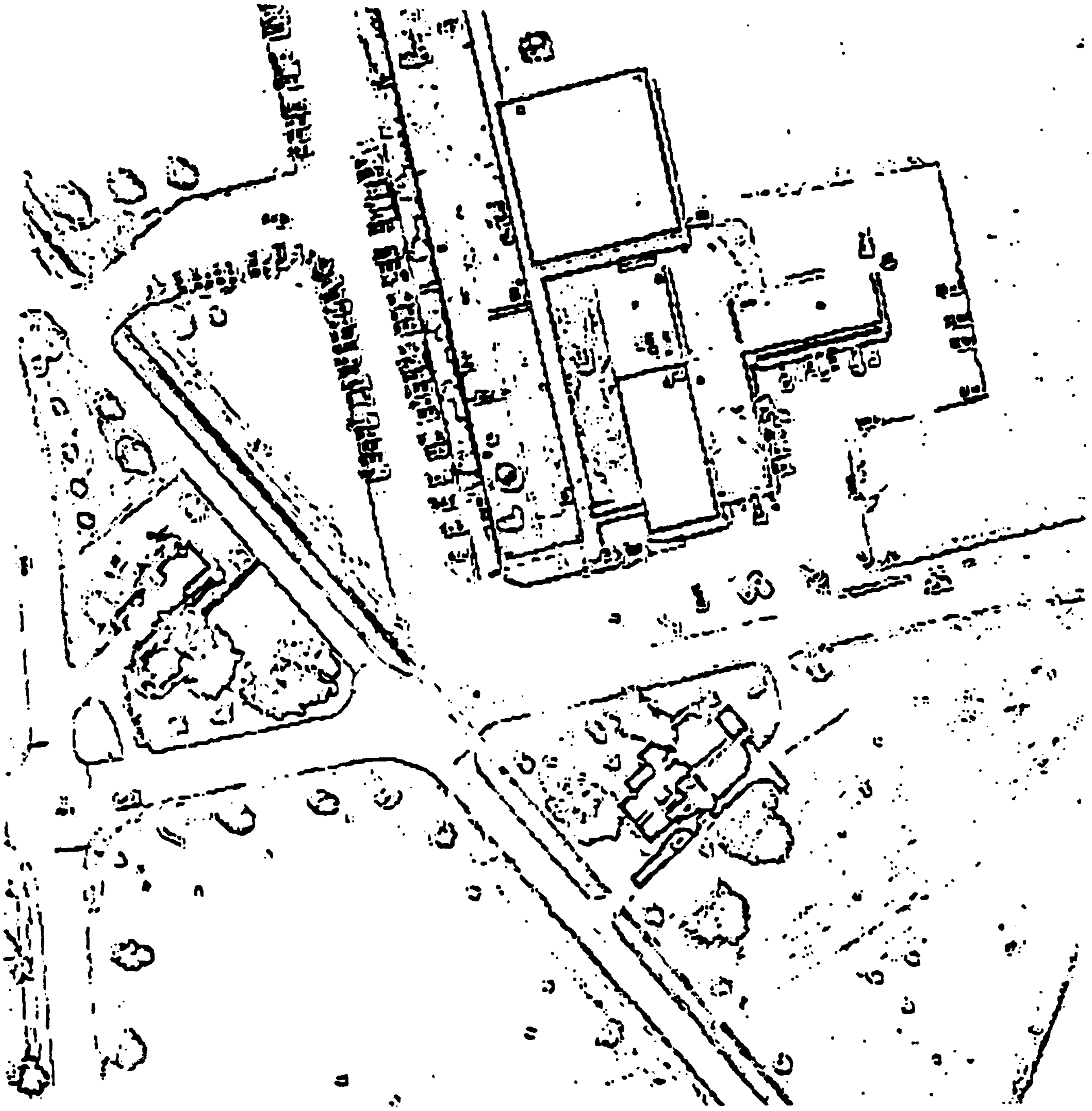


Figure 4-19b. A thresholded edge strength image with bright pixels only.

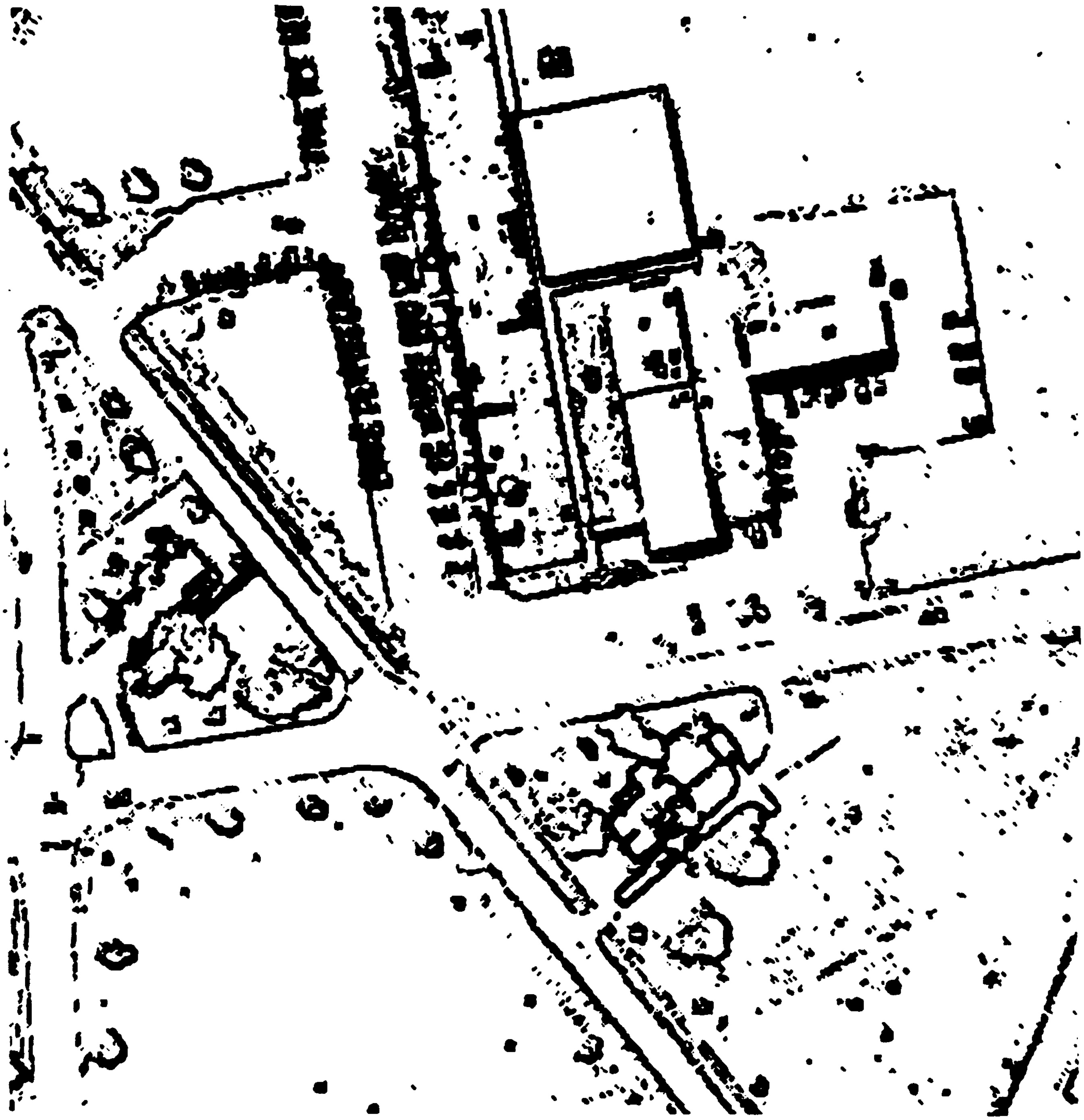


Figure 4-19c. A thresholded edge strength image with all bright and dark pixels.

4.7.2 Selective Seeding

Suppose a known seed is implanted in a uniform area with an unknown size. If the seed starts to grow within this area at the end all pixels in the area are marked with the seed identifier. In this case, by checking the mark, the size of this area can be precisely measured. The same concept is adopted for the OB object detection.

First, seeds for cars must be found. They are then implanted in the edge strength image obtained as described in 4.7.1. After proper expansion, cars are extracted and counted. Since the image contains many other objects, several selective operations are also utilized to clean out unwanted large or small objects. The implementation of this concept is defined in the following steps.

Step 1: Extract seeds for cars.

1) Apply bright object and dark object detection algorithms described in section 4.1.3. Notice that most of the bright pixels in a car and dark pixels in a dark car are detected during this process. In addition, they also form many small disconnected regions. These are excellent seeds for the car detection. However, they are accompanied by many other large and small objects. Since car seeds are small objects, by checking the object's size, unwanted large objects can be easily eliminated. The following steps are designed to achieve this purpose.

2) Do labelling to both bright object and dark object images.

3) Extract object size from both labelled images.

4) Compare it with the expected object size, and keep all objects with a size smaller than or compatible with the expected object size.

5) Combine small objects detected from both images into a single seed image.

At the end of 5), seeds for small objects including cars are obtained.

Step 2: Implant and Grow.

Seeds are implanted into the edge strength image produced as described in section 4.7.1. The expansion process uses the object boundary in the edge strength image as the limit, and grows the boundary to a size compatible to that of the expected object size (i.e., iterates a fixed number of times). Over-growing is harmful and should be avoided. Since the edge strength image contains mainly

boundary pixels, the technique described in 4.1.5 shall be employed to fill in the object body.

Step 3: Car detection and counting.

1) Perform the labelling technique on the last processed image.

2) Extract object size.

3) Check the size to make sure that it is wide and long enough to be considered one or more of the objects of interest.

4) Count the total number of pixels in the object.

5) Compare it with the total number of pixels expected for the desired object.

6) Decide the number of instances of the OB object which are present in the region under consideration, and accumulate a total count for the area being processed.

It is clear that if in the image there are other dark or bright objects of the same size as the desired OB object, they are likely to be counted as instances of the desired object. Thus proper definition of the search area is important for the algorithm in it's current state.

4.4 New Image Camera Model

The FTCA environment provides for the capability to compute the camera model for a new mission image collected for a target area for which a site model exists. All of the database information and exploitation techniques can be applied to the new mission data once the camera model has been defined. Future collection systems are expected to convey sensor parameters along with the ephemeris data such as to obviate the necessity for building camera models for each new mission image. Once the camera model is available, the site model will correctly project over the new mission image data.

4.4.1 New Camera Model Process Overview

The process by which the camera model is built for the new mission image requires the selection of conjugate points between objects within the site model and the mission image for which the new camera model is desired. In order to build the new camera model approximately eight to ten conjugate points must be selected. The existence of a site model helps in the process of picking reference data points.

Following the selection of approximately eight to ten conjugate points, the user can initiate the building of a camera model for the new mission image. This mission image camera model can be subsequently refined until a highly accurate transformation between the site model and new mission image is achieved as determined by the degree to which the wire frame projection of the site model aligns with mission image objects.

The process for building a new camera model can be achieved on a single GLMX stack or two GLMX stacks may be employed. The following discussion will suppose the construction of a camera model using two GLMX stacks. One GLMX stack should display the reference image, while the second GLMX stack should display the mission image. Each GLMX stack should include the site model wire frame projection process (i.e., database) such that the target area site model may be projected over the reference image initially and subsequently over the mission image as well. Since no camera model exists for the mission data when the process begins it will not be possible to project the target area site model over the mission image.

The first step in creating a new camera model will be to select the "New Camera" option under the "Utility" menu. The user may add the process to the mission data GLMX stack or the reference data GLMX stack when the new process cue appears on the screen. When the process is added to the GLMX stack a new tab, named "Picker", will appear on that stack. The user should then clone the "Picker"tab using the "Clone" menu option on the "Picker" Tab application menu, and attach the cloned process to the remaining GLMX stack, such that the "Picker" function tab shows on both the reference and the mission image GLMX stacks . With both "Picker" processes popped to the top of the stack the user may proceed in selecting conjugate match points.

In addition to the appearance of the "Picker" tabs a third window, the "Camera File" application window, will appear on the display screen which will allow the user to monitor the selection of conjugate point pairs as they are produced for the new image. Conjugate points may be added to or deleted from this window as the process proceeds. Also, when the "Picker" function is at the top of the GLMX stack and the cursor is in the GLMX window content area, the cursor will form a cross hair sight to facilitate point selection by the operator. Point pairs are generated by surveying the new mission image data to identify features on which clearly define points can be selected. These should well defined points such as corners of buildings, corners of modeled parking lot areas, etc. Rooftops typically make very convenient features from which to extract multiple data points. An operator should initially inspect the mission and reference image data in order to establish the relative orientation between the two scenes. This information will allow the operator to pick individual points that correspond on a given structure.

A handy feature of the "Picker" function is the direct zoom capability which assists in the accurate placement of point selections over the mission image. By placing the cursor over the general area on which a point is to be selected and by pressing the middle mouse button, the image is automatically zoomed to a level that allows accurate point identification. A second click of the middle mouse button will cause the image to zoom out to its "Show All" view.

In order to generate a matched pair of conjugate points the operator should point to a particular feature (say a corner) of an object in the site model in either the mission or reference image window. Once the mouse cursor is placed as accurately as possible over that particular feature the left mouse button should be pushed. A green marker will be drawn over the selected point. The green color of the marker indicates that the point represents a marker of an uncompleted conjugate pair. The operator should then move the cursor to the corresponding point in the opposite image window and press the left mouse button again when the cursor is placed as accurately as possible over the corresponding point. When the conjugate point is selected the green color of the original marker changes from green to yellow and a conjugate pair entry is made in the "Camera File Application" window.

The point selected on the reference image will automatically be placed at the closest vertex of the site model object. This facilitates the selection of points accurately on the reference image as a selected point will automatically be attracted to a corner vertex of the site model object. It is best to pick the reference image point first, to avoid the generation of an unintended conjugate pair. This event can occur if the marker for the second point being selected in a conjugate pair is attracted to a close, but erroneous, vertex of the site model object overlaying the reference image. If errors are made the deletion of such pairs is a simple matter, accomplished by the "Delete Pair" option under the "Picker" application menu. The first point of an uncompleted conjugate pair can be re-picked until satisfied with it's placement; thus if the marker is unintentionally attracted to an erroneous "close" vertex on a site model object vertex, a simple repositioning of the mouse and re-click can remedy the problem.

Once an operator has established a single conjugate pair on a structure such as a rooftop, it is a simple matter to identify other corresponding points on that same rooftop by following the clockwise rule. The clockwise rule is simply that a clockwise adjacent point (e.g. on a roof surface) to a known conjugate point on one image will be conjugate to the point that is clockwise adjacent to the initial referenced point conjugate on the other image. This rule allows one to establish a single conjugate point on a rooftop, for example, and then select subsequent point pairings by sequentially identifying clockwise adjacent points to the initial conjugate

pair on each rooftop. Using the technique eight or nine conjugate points can be quickly established between the mission and site model vertices.

If four or five points are located on a given structure, and another four or five points are located on another structure which is widely separated within the target area, a camera model can usually be built and generated from those points which is accurate enough for initial recognition and used as an interim camera model to assist in the selection of subsequent refinement points. To generate the interim camera model move the cursor to the "Camera File Application" window and press the right mouse button, displaying the "Build Camera Model" menu option. When the camera model building process is finished move the cursor to the mission image GLMX stack and pop the mission image tab to the top of the stack. From the mission image application menu Select the "New Camera" option with the right mouse button, from the mission image application menu. These actions will retrieve the new mission image camera model.

The target area site model, using the interim mission image camera model, may be displayed over the mission image by turning on the "Data Base" tab over the mission image. The wireframe depiction of the site model will appear over the mission image but may be more or less distorted depending on the accuracy and position of the trial data points that have been selected. Generally the identification of one or two more conjugate points in the vicinity of the target area where maximum model error appears is sufficient to bring the model quite closely into alignment.

If further work on camera model development is desired to perfect the camera model, the subsequent process can be facilitated through the use of the locking function (defined in section 3.2.5.2) between windows. The locking process eliminates the need for the operator to decide which points on a building structure are conjugate since the locking function causes the immediate display of conjugate points on the two images. Moving to points of poor database alignment and using the middle mouse key in the point selection mode to zoom on those points immediately brings up the conjugate point on the mission image.

4.5 Image Alignment Via Site Model

The FTCA System includes a 3D-based image alignment function. We define this registration function as follows: Registration is the result of the transformation of one of two pictures of a nearly common scene so that the transformed picture (image) and the other (original mission) are nearly spatially identical. It is also possible that registration could be achieved by transforming each image so that the two transformed pictures are nearly identical, however this latter approach is not

used by the FTCA alignment process. It is also possible that one or both of the images may be synthetic, i.e., generated from a digital database.

Three general statements can be made concerning the transformation necessary for registration:

1) It is dependent on the scene's geometry.

2) In general, it does not exist for all points.

3) In general, it is discontinuous even if the scene is a continuous surface.

These generalities are illustrated in Figure 4-20a, b, and c. Shown here are one-dimensional images of two-dimensional scenes. They also are true for the case of interest, i.e., two-dimensional images of three-dimensional scenes.

Registration is merely an intermediate step in the FTCA process. However, it is easier for automatic systems and humans to find changes if the two images are registered. The better the registration, the easier it is to detect changes. But, since the registering transform is scene dependent, and a complete description of the scene is seldom given, registration will always be somewhat imperfect.

4.5.1 Functional Form of Transformation

Let (u,v) be the coordinates of a point on the first image and let (x,y) be the coordinates of its match point on the second image. Then the registration transform is

$$\tilde{x} = f(u,v) \quad , \quad \tilde{x} \approx x$$
$$\tilde{y} = g(u,v) \quad , \quad \tilde{y} \approx y$$

if the first image is transformed into the second. Or, it may be written as

$$\tilde{x} = f_1(u, v)$$
$$\tilde{y} = g_1(u, v)$$
$$\tilde{u} = f_2(x, y) \quad , \quad \tilde{x} \approx \tilde{u}$$
$$\tilde{v} = g_2(x, y) \quad , \quad \tilde{y} \approx \tilde{v}$$

if each image is transformed so that the new images are nearly spatially identical.

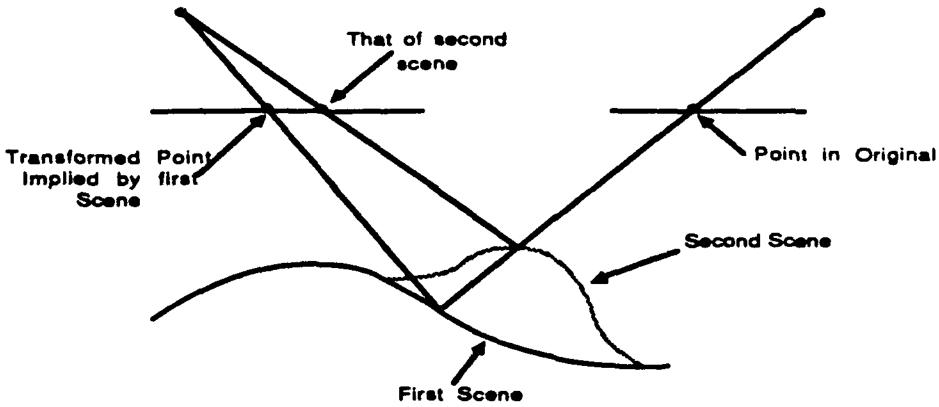


Figure 4-20a. Scene dependence

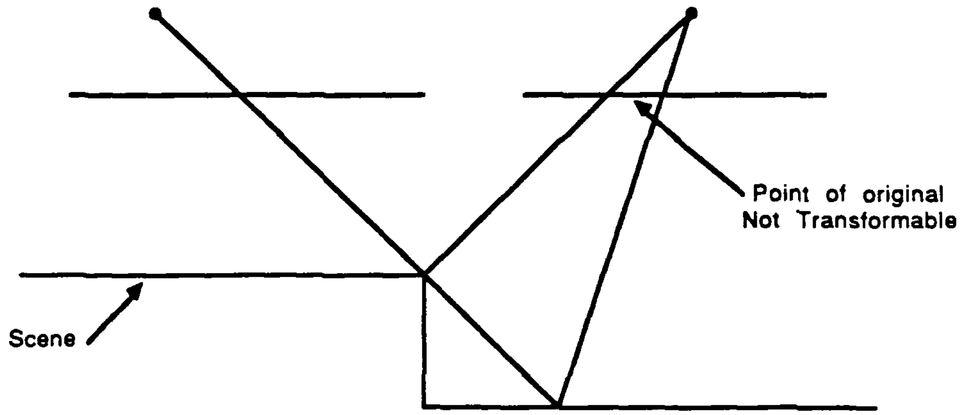


Figure 4-20b. Non-Existence of certain points

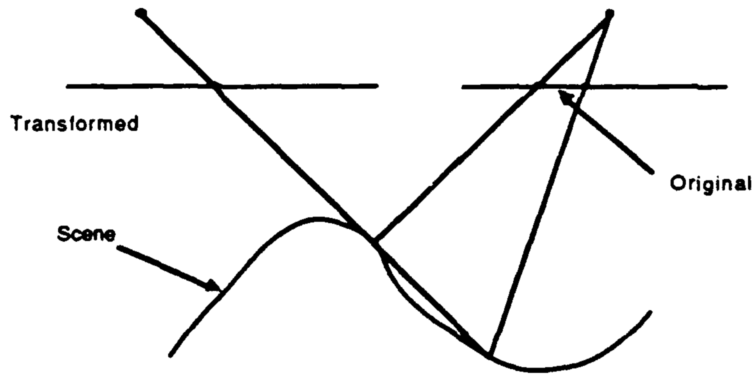


Figure 4-20c. May be Discontinuous
Figure 4-20. Illustrations of three anomalies which may exist in transformations of one image onto another.

Development of the functional form of the transformation is provided by changing from a given image to a new image such that the new image is formed with position and attitude different than that of the given image. This is illustrated in Figures 4-20 and 4-21. The principle is simple. Choose a point on the original. The preimage of this point is a ray in three-space. Determine the point at which this ray pierces the scene. This scene point may or may not be visible from the new camera position as shown in Figure 4-20b. If it is visible place the intensity of the chosen point at the image of the scene point under the new position and attitude. Errors in this transformation are incurred if the camera models of the two original pictures are inaccurate or if the description of the scene specification is inaccurate.

If the camera model is that of a central perspective and the scene is planar (or composed of a number of planar surfaces) then the transformation may be written

$$f(u, v) = \tilde{x} = \frac{K_1 \bar{x}}{K_3 \bar{x}}$$

$$g(u, v) = \tilde{y} = \frac{K_2 \bar{x}}{K_3 \bar{x}}$$

where $\bar{x} = (u, v, 1)^T$

$$K = A\left[\left(d - \bar{N}_1^T \hat{n}\right) I - \left(\bar{N}_2 - \bar{N}_1\right)\hat{n}^T\right]^{-1} B^T,$$

A, B are the original and transformed camera attitude matrices respectively,

$\bar{N}_1, \bar{N}_2$ original and transformed camera position respectively

$\hat{n}$, d unit normal and distance from three-space origin of planar scene.

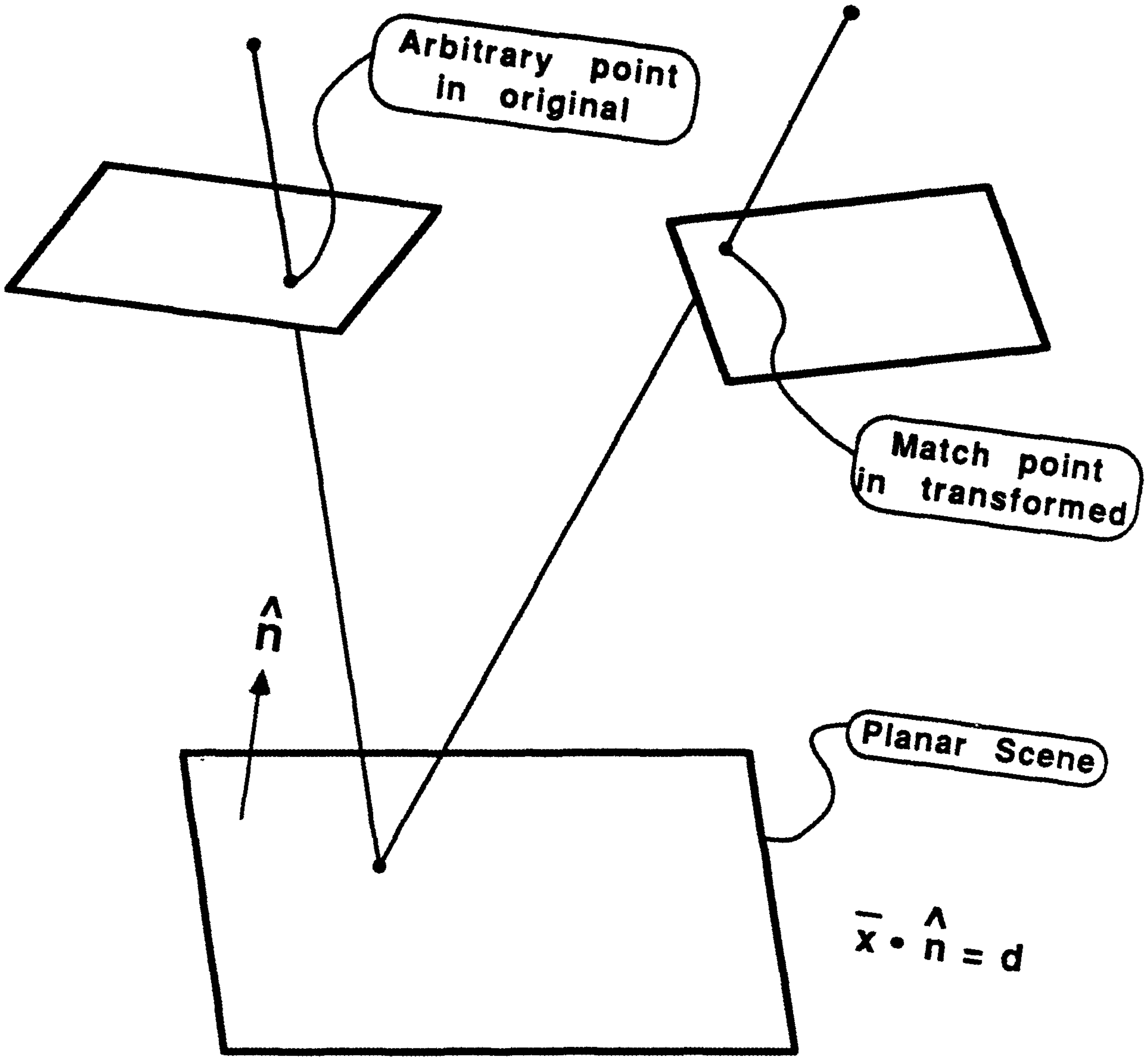


Figure 4-21. Functional form of transformation for planar scene.

Note that the transformation is a ratio of linear terms. Some authors mistakenly use a linear transform and then ponder the reason for the resulting large errors.

5.0 TARGET AREA PROCESSING RESULTS

This section will provide a pictorial review of the FTCA System presentation options and provide some insight into the system's functionality as well. As always this is difficult to convey in any set of static images. For readers interested in the system's capabilities and performance, a visit to Rome Air Development Center would be more informative, especially in view of security considerations on the imagery used for testing purposes. The images used for the discussion below are unclassified.

5.1 Target Area Source Imagery

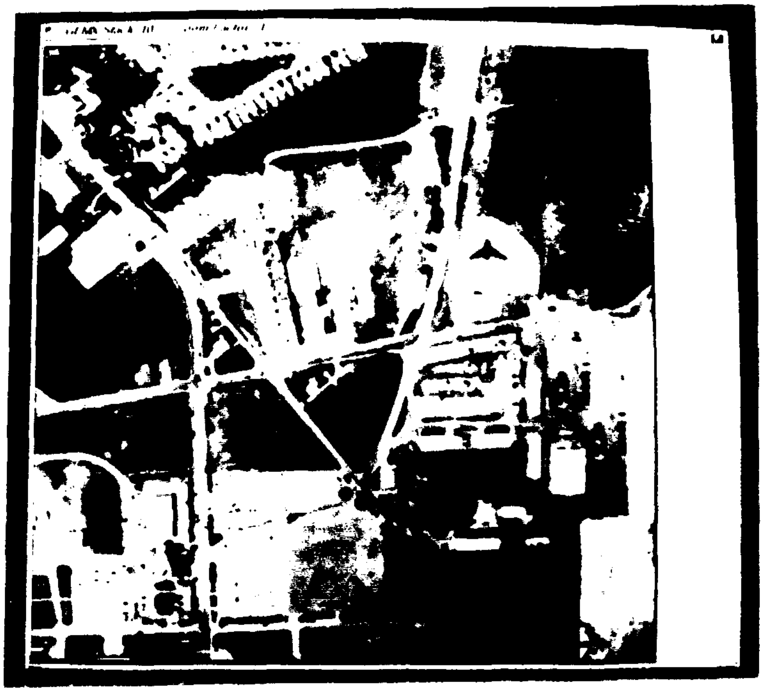
Figure 5-1a, b, and c are 3 images over our hypothetical target area. Figures 5-1a and b were taken via aerial photography. No particulars on the camera used or sensor position at the instant of exposure are known. Figure 5-1c is believed to be an aerial photograph using IR film taken at least 10 years prior to the first two images. Once again no collection information is available for this image.

5.2 Target Area Preview

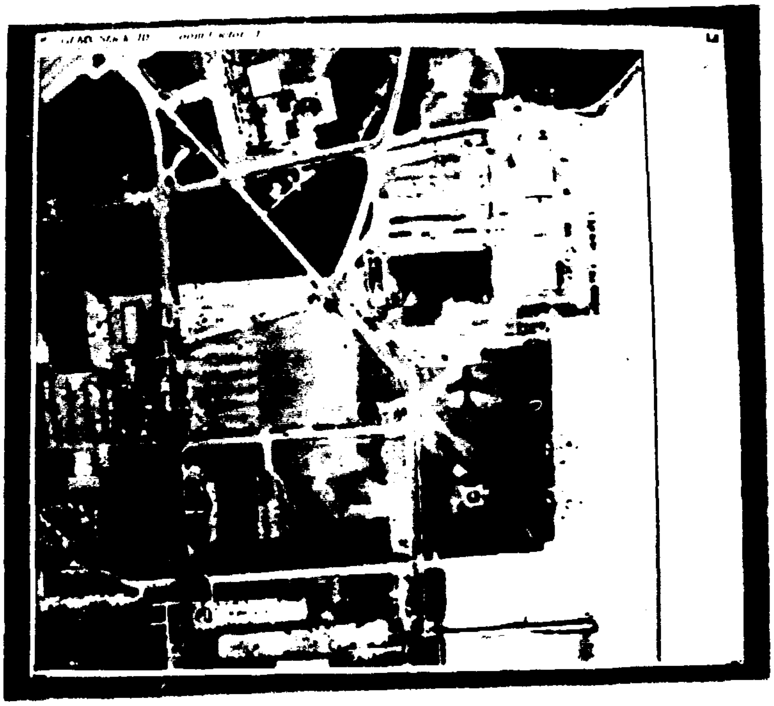
One of the key steps in reading out a target area is for the Image Analyst to get familiar with the content and important features of the target area. If the area is read out every day by the same analyst, this problem is minimized. If on the other hand, the target area is read out infrequently or by different analysts, this can take much more time. One of the innovative aspects of the FTCA System is it's incorporation and usage of a 3 dimensional target area site model. The existence of this site model provides several meaningful approaches for improved Image Analyst previewing of the target area.

The FTCA system provides multiple target area preview options to address the issue of target area familiarization. Figure 5-2a and b depict computer generated images of the target area 3D site model. The analyst has the option to change viewing position in real time through mouse button selections to allow viewing of the target area from any perspective.

Figure 5-3a, b and c show an additional preview mode in which wire frame depictions of the site model are overlaid on the reference and mission images for the analyst. A preview application allows the analyst to step through the objects in the site model one at a time providing a pointer and object identification. Optionally close-up and automatic stepping modes can be selected. Using GLMX services, the analyst can select the preview overlay on either image at any time. This provides a rapid preview and familiarization capability for the Image Analyst on multiple images.



a. Target area source image 1.



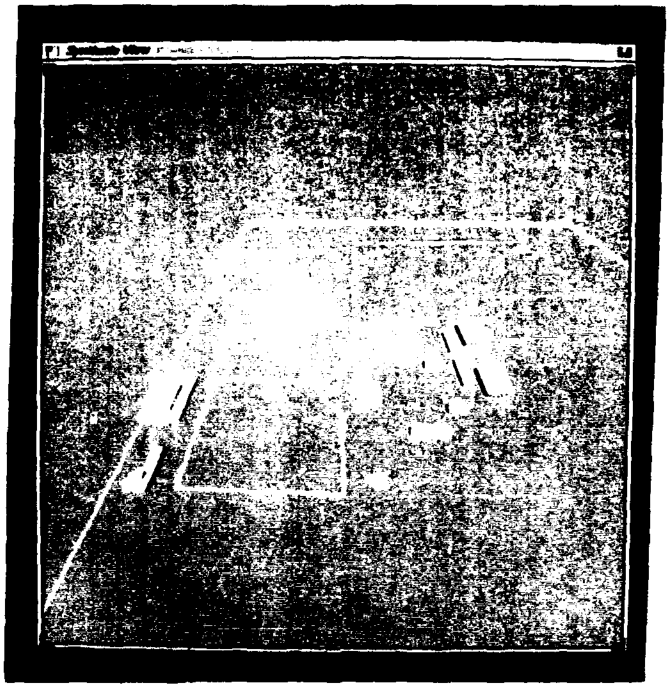
b. Target area source image 2.

c. Target area source image 3.
Figure 5-1. Target area source images.

An additional preview mode is provided which includes a visual note taking and information sharing facility. Target area annotation data (e.g., text, graphics, etc) can be generated interactively by the analyst, stored on a file for the target area, and recalled and projected as overlays on any of the images covering the target area at any time. This is possible since the annotation data is stored in a 3 dimensional coordinate system tied to the site model. Figure 5-4a and b illustrate the capability. Thus Image Analysts can leave visual records of their finding to share with the next analyst or to cue themselves for the next time they want to read out the target area.

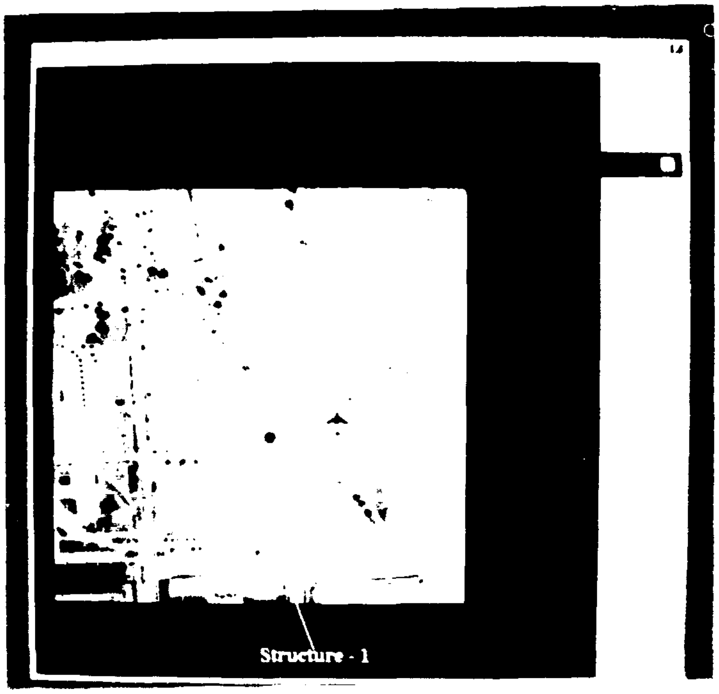
5.3 Automated Comparative Analysis

The FTCA System provides multiple techniques for performing automated comparative analysis. These techniques were discussed in section 4.2. The techniques can be all or selectively applied to a target area. All of the techniques are run as background tasks. For the target area under discussion, on the delivered workstation, approximately 45 minutes of processing time would be required for all techniques to be run. After processing is complete, the Image Analyst can request a review of the processing results. Figures 5-5a, and b in conjunction with 5-6a and b, attempt to show how the system provides that summary to the Image Analyst.



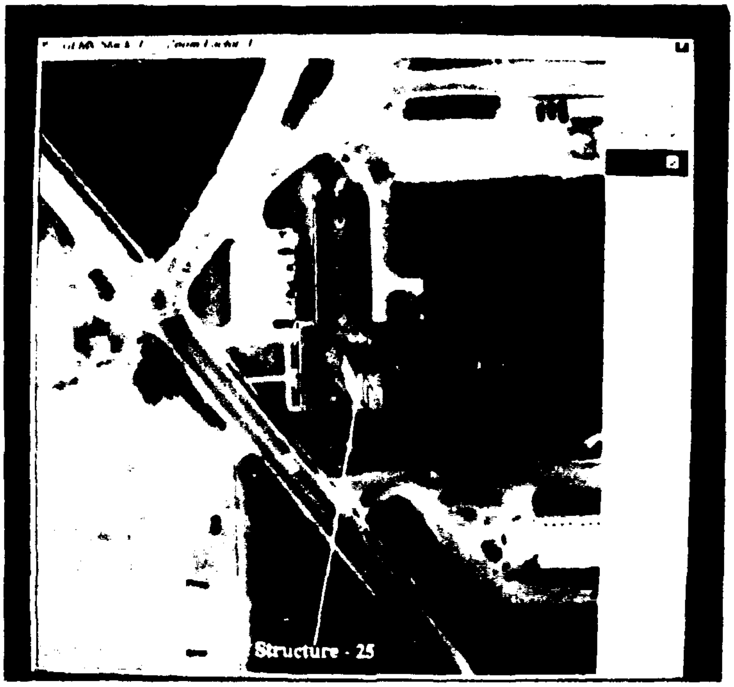
a. Synthetic target area image.

b. New Perspective and Scale.
Figure 5-2. Synthetic Target area images.



a. Site model preview image 1.

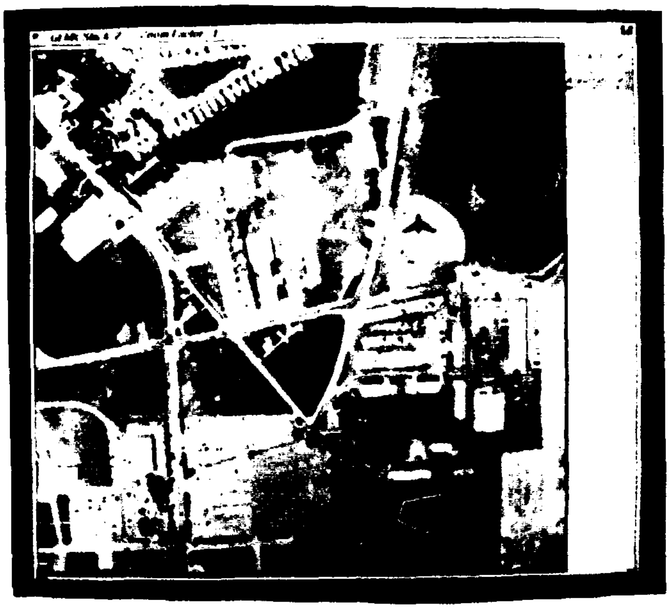
b. Site model preview image 2.



c. Site model preview close-up.
Figure 5-3. Target area Preview.

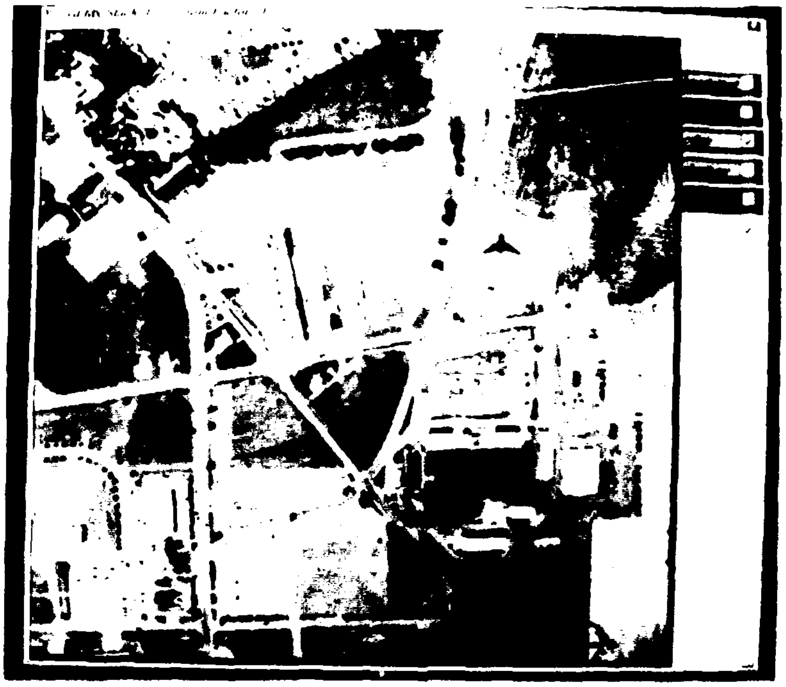
As part of the comparative analysis, an aligned reference image is produced which spatially and radiometrically matches the current mission image. The alignment process relies upon knowledge in the site model to very precisely map the reference intensities as if collected from the mission image perspective. Figure 5-5a shows a mission image and 5-5b shows the aligned reference image produced by the system. Each window on the system is really a double buffer so these two images can be viewed in a flicker mode by copying either to the back buffer, selecting the opposite for viewing, and requesting blink. This in itself is a valuable aid for the Image Analyst since even subtle change areas appear to "pop out".

Each of the comparative analysis techniques produce a "results file". The Image Analyst can request a presentation of the processing results in which the results from each of the various techniques is available for comparing and contrasting. Figures 5-6a and b provide an example. Notice that a color-coded GLMX information tab is provided for controlling the display of results from each of the comparative analysis techniques. The Image Analyst can simultaneously view any or all of the cues produced by the techniques on either of the 2 images. The cues can be selectively changed between outline and opaque-fill cueing options. Figure 5-5d illustrates that the focus area, content, and zoom factors can be interactively changed during this review process.



a. Annotation overlay on image 1.

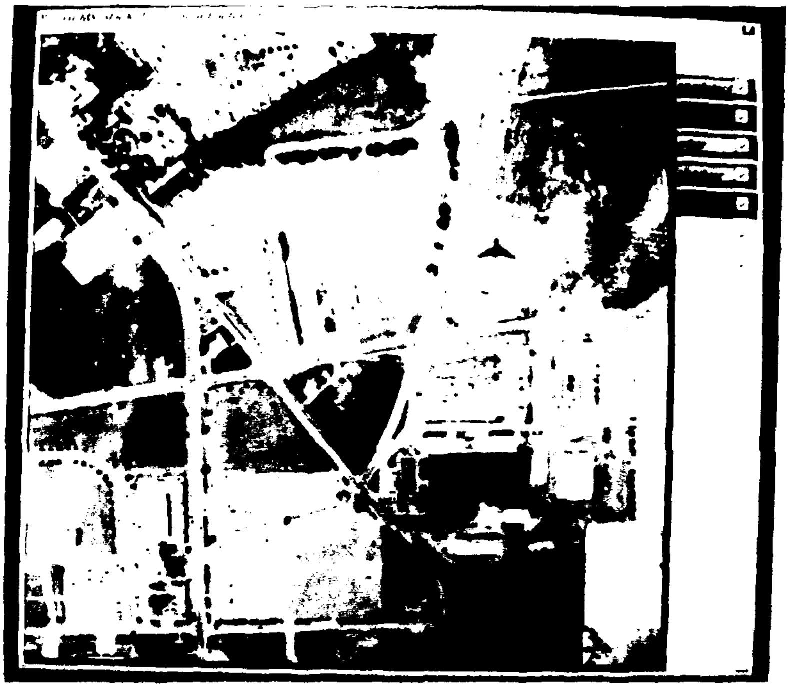
b. Annotation overlay on image 2.
Figure 5-4. Target area annotation.



a. Mission image.

b. Aligned reference image.

Figure 5-5. Image alignment for comparative analysis.



a. Automated comparative analysis cues.

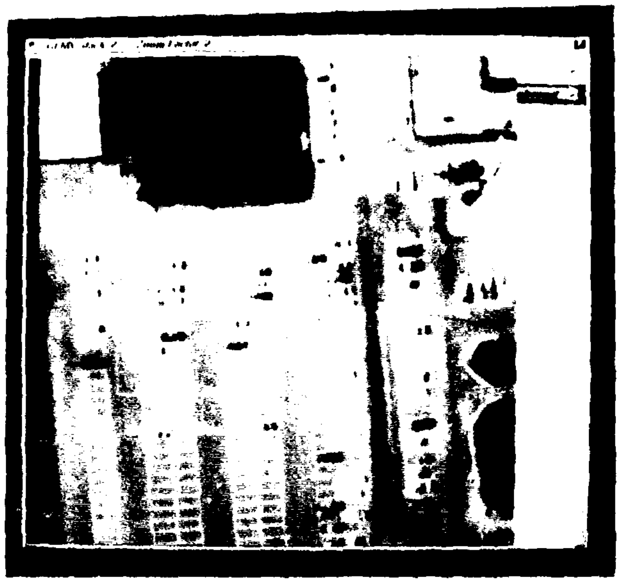
b. Zoom on selected area and information.

Figure 5-6. Automated comparative analysis results presentation.

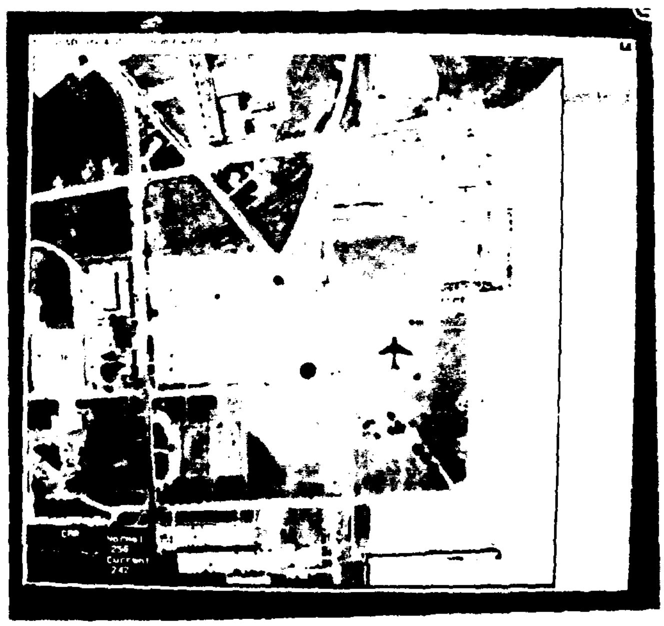
5.4 Order of Battle Object Detection and Counting

The FTCA system also includes an OB detection and counting function. This function can be applied in two ways. The first of these is an interactive, user-directed facility for selectively picking object type and search area as described in section 3.2.3.4. When applied in this fashion the system responds with a presentation as illustrated in figure 5-7a. The count is provided as well as an opaque-filled mask over the detected objects.

This function can also be applied to predefined regions over the target area. An interactive procedure for defining "count areas" is provided. If the "count all" option is selected the system processes each area and saves the result. The Image Analyst can request to review the results of counting and the results will be presented as illustrated in figure 5-7b. The areas are outlined, individual color-coded counts are provided, and a graphical summary for the entire target area is also provided for facilitating trend analysis and to detect deviation from normalcy.



a. OB count over selected area.



b. OB count results over designated areas.

Figure 5-7. OB detect and count function presentations.

6.0 CONCLUSION AND RECOMMENDATIONS

This final section summarizes our findings and offers recommendations and ideas for additional studies and further development in light of the results of this contractual effort.

6.1 Conclusions Summary

As a result of this development effort we offer the following as major summary conclusions:

- 3D model-supported exploitation approach was correct choice for FTCA
- the GLMX Windowing System is well-suited to the requirements
- the delivered FTCA baseline is easily extensible.

Having a site model to support exploitation is clearly a significant advantage. Cognizance of the expected target area objects allows information extraction algorithms to be be applied efficiently and intelligently. Rapid generation of site models is required for the model-supported exploitation concept to become reality. Efforts are underway in this area in a number of different organizations indicating this will become a reality in the near term. Thus the decision to pursue a 3D model-supported exploitation concept was and remains the right decision.

The GLMX Windowing System appears to be well suited to the image exploitation task. GLMX's ability to utilize the site model coordinate system to fuse information over multiple images and other sources of information provides an intuitively obvious, user friendly environment for the analyst. The ability to effectively "stack" information in a single display window allows efficient use of limited display resource. Allowing the analyst to select/deselect any combination of information sources provides the analyst with the ability to program the amount and order of information presented as the exploitation job proceeds.

The system as delivered offers a uniquely extensible environment. Each of the image analyst's tools currently available in the FTCA system were independently developed with it's own user interface. By utilizing the provided GLMX programming resources and standards each of the applications can be "clipped" into the GLMX environment. Additional automated exploitation techniques and analyst's tools can likewise be produced and easily added to the baseline capability.

6.2 Recommendations Summary

As a result of this development effort we offer the following as recommendations and/or ideas for additional development:

- more study/testing/refinement of existing techniques required
- additional techniques should be developed
- automatic camera model generation should be investigated
- process more target types
- multi-sensor processing/fusion
- site evaluations of provided Image Analyst tools
- interim hardcopy product generation for evaluation
- interface directly with site model production/refinement

The FTCA development like many other concept development efforts strived to provide as much breadth as possible in the range of tools and techniques for consideration. The delivered capability needs to be more thoroughly tested and undoubtedly refined. A substantial amount of time could be spent in this area.

Additional automated exploitation techniques should be developed to take advantage of the existence of a site model. As previously discussed, techniques to make better use of surface material data should be developed. In addition, better collective use should be made of the results of each of the individual comparative analysis techniques. Initially this could be as simple as prioritizing the cues presented to the image analyst based upon a simple voting scheme. The baseline image analyst toolset should be extended. Using the 3D presentation and manipulation capabilities of the workstation, a visual interpreter keys capability could easily be added. In addition, the preview function should be extended to allow text descriptions of site model objects to be selectively presented.

FTCA provides a capability for defining a camera model for a new mission image given the availability of a site model. This allows new images to be brought into the exploitation environment. Essentially this is an operator-directed approach in which a number of corresponding points between the site model vertices and mission image locations must be identified. We believe this process is a good candidate for automation. It is clearly a critical requirement for making automated exploitation a reality, since many new mission images will need to be examined in a production environment. In addition, the existence of a site model, provides a significant advantage in terms of deciding what features to attempt to locate in the mission image. Finally in a production environment, we would expect that the sensor pointing data would accompany the new image, and at a minimum would provide an initial close approximation to the image's camera model which would then need only refinement.

The FTCA system was developed and tested for a limited number of fixed target area types. Clearly there are many more of interest. More should be explored both from a site-modeling and subsequent automated exploitation point of view.

The FTCA model-supported exploitation approach offers some interesting possibilities in the area of sensor image data fusion. Interpreting image data from multiple sensors over cultural feature objects can be very difficult for some types of sensors. The use of site models would make this might easier if the right set of interpretation aids were developed. Exploiting data from more sensor types should be undertaken to investigate the possibilities.

The FTCA system was intended for an installation like the 480th RTG/INPOE at Langley Air Force Base. It should be very informative to have personnel at selected sites attempt to use the system for a typical target area readout over a reasonable time interval. Obviously the logistics of softcopy exploitation at these sites would need to be worked out, but the insights and observations gleaned from everyday users would be extremely useful.

It would also be interesting to attempt to evaluate the potential for model-supported exploitation in a more restricted sense. Most installations exploiting imagery today are essentially hardcopy based. It would be informative to prepare target folders with some hardcopy products which could easily be produced by the FTCA workstation and then evaluate their utility by monitoring how often they are used by Image Analysts. Candidate products might include imagery with wire frame overlays, solid model views from multiple aspects, annotated imagery, or images generated from selected viewer positions (image perspective transformation). Once again the details of deciding what products might be of use and the procedure for emulating the ease of softcopy updates would need to be worked out.

The model-supported exploitation functions available within FTCA obviously rely on site models. The software expects site models to be provided in a particular format. Some consideration should be given to providing FTCA with site models from other available modeling packages. We anticipate that the ARSP system, which will provide site modeling capability, and FTCA should be compatible until any potential changes in site model data bases formats driven by new site modeling requirements are defined. The potential for accepting site models from other software packages should be investigated as well.

7.0 REFERENCES

[1] A. P. Reeves, M. L. Akey, and O. R. Mitchell, " A Moment-Based Two-Dimensional Edge Operator", Proc. IEEE Comput. Soc. Symp. Computer Vision and Pattern Recognition, June 19-23, 1983.

[2] E. P. Lyvers, and O. R. Mitchell, "Precision Edge Contrast and Orientation Estimation", IEEE Trans. Pattern Analysis and Machine Intell., Vol. 10, pp. 927-937, Nov. 1988.

[3] R. Park, and W. Y. Choi, "A New Interpretation of the Compass Gradient Edge Operators", Comput. Vision, Graphic, and Image Proc., Vol. 47, pp. 259-265, 1989.

[4] M. Nadler, "A Note on the Coefficients of Compass Mask Convolutions", Comput. Vision, Graphic, and Image Proc., Vol. 51, PP. 96-101, 1990.

[5] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A Survey of Thresholding Techniques", Comput. Vision, Graphic, and Image Proc., Vol. 41, pp. 233-260, 1988.

[6] S. S. Reddi, S. F. Rudin, and H. R. Keshavan, "An Optimal Multiple Threshold Scheme for Image Segmentation", IEEE Trans. Systems, Man, and Cybern., Vol. SMC-14, pp. 661-665, 1984.

[7] Flavio R. Dias Velasco, "Thresholding Using ISODATA Clustering Algorithm", IEEE Trans. Systems, Man, and Cybern., Vol. SMC-10, pp. 771-774, 1980.

[8] N. Ostu, "A Threshold Selection Method from Gray-Level Histogram", IEEE Trans. Systems, Man, and Cybern., Vol. SMC-8, pp. 62-66, 1978.

[9] T. W. Ridler, and S. Calvard, "Picture Thresholding Using an Iterative Selection Method", IEEE Trans. Systems, Man, and Cybern., Vol. SMC-8, pp. 630-632, 1978.

[10] D. C. C. Wang, A. H. Vagnucci, and C. C. Li, "Digital Image Enhancement: A Survey", Comput. Vision, Graphic, and Image Proc., Vol. 24, pp. 363-381, 1983.

[11] G. Borgefors, "Distance Transformations in Arbitray Dimensions", Comput. Vision, Graphic, and Image Proc., Vol. 27, pp. 321-345, 1984.

[12] B. Bhanu,"Automatic Target Recognition: State of the Art Survey", IEEE Trans. Aerospace and Electronic Systems, Vol. AES-22, pp. 364-379, 1986.

[13] M. Kunt,"Edge Detection: A Tutorial Review", Proc. IEEE Comput. Soc. Symp. Computer Vision and Pattern Recognition, July 1982.

[14] Richard Lippmann, "An Introduction to Computing with Neural Nets," IEEE ASSP Magazine, April, 1987.

[15] J. L. McClelland and D.E. Rumelhart, Explorations in Parrallel Distributed Processing, The MIT Press, Cambridge, 1988.

[16] P.E. Gill and W. Murray, "Conjugate-Gradient Methods for Large-Scale Nonlinear Optimization" System Optimization Lab, Standard University, Tech Report SOL 79-15, Oct., 1979.

[17] H.R. Paetznick, C.B. Grosch, "Geometric Surface Shell Construction Process", Final Technical Report, RADC Contract F30602-81-C-0103, Control Data Corporation, Sep.,1984.

MISSION

OF

ROME LABORATORY

Rome Laboratory plans and executes an interdisciplinary program in research, development, test, and technology transition in support of Air Force Command, Control, Communications and Intelligence (C^3I) activities for all Air Force platforms. It also executes selected acquisition programs in several areas of expertise. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C^3I systems. In addition, Rome Laboratory's technology supports other AFSC Product Divisions, the Air Force user community, and other DOD and non-DOD agencies. Rome Laboratory maintains technical competence and research programs in areas including, but not limited to, communications, command and control, battle management, intelligence information processing, computational sciences and software producibility, wide area surveillance/sensors, signal processing, solid state sciences, photonics, electromagnetic technology, superconductivity, and electronic reliability/maintainability and testability.